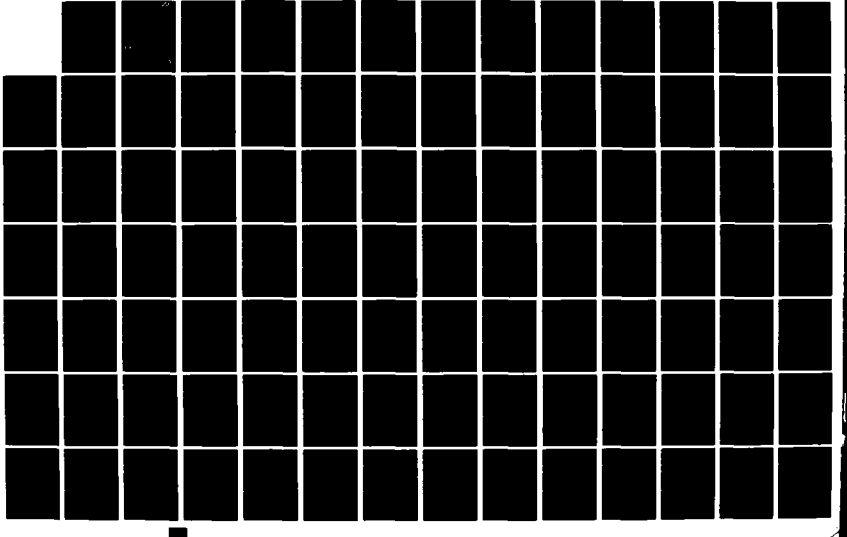
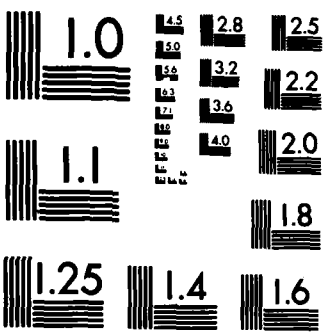
AD-A122 336 RPSL1D (A ONE-DIMENSIONAL VERSION OF REPSIL)(U) ARMY 1/2
ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERDEEN
PROVI... J D WORTMAN NOV 82 ARBRL-MR-03221

UNCLASSIFIED SBI-AD-F300 132 F/G 19/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Memorandum Report ARBRL-MR-03221

RPSL1D
(A ONE-DIMENSIONAL VERSION OF REPSIL)

John D. Wortman

November 1982

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Memorandum Report ARBRL-MR- 03221	2. GOVT ACCESSION NO. AD-A122336	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* RPSL1D (A ONE-DIMENSIONAL VERSION OF REPSIL)		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John D. Wortman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Ballistic Research Laboratory ATTN: DRDAR-BLT Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N/A
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Research & Development Command US Army Ballistic Research Laboratory (DRDAR-BL) Aberdeen Proving Ground, MD 21005		12. REPORT DATE November 1982 13. NUMBER OF PAGES 147
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* Director US Army Materiel Systems Analysis Activity Aberdeen Proving Ground, MD 21005		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*
Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Structural Response Code	Elas[illegible]lastic Behavior
Finite Difference Method	Imp[illegible]ressure Loading
Computer Program	Axial [illegible]tric Shells
Large Deformation	Slab Symmetric Shells
Thin Kirchhoff Shells	Laterally Symmetric Beams

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

RPSL1D is a one-dimensional version of REPSIL, a computer program which calculates the large plastic deformation of thin Kirchhoff shells. The REPSL1D program is more efficient than REPSIL for axially symmetric shells (shells of revolution) and slab symmetric shells. It has been used most frequently for deformation calculations in laterally symmetric beams. The numerical algorithm differs from that of REPSIL as recorded in Reference 2 in that external, artificial, points are used to define boundary conditions,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

and that forces are computed between mesh points as well as at mesh points. This gives a better simulation of clamped ends and a more compact, more stable system of difference equations. The report gives instructions for using RPSL1D. Listings of the program and the companion plotting program are given. Two example problems are discussed with tabulated input and output. A number of options, primarily for beams, are tabulated and discussed.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

1. INTRODUCTION

RPSL1D is a spin off of the FORTRAN IV computer program REPSIL[1,2] developed at the Ballistic Research Laboratory to treat large, transient, elastoplastic deformations in thin shells. Thin shell programs, such as REPSIL, replace the essentially three-dimensional geometry of the deforming shell with a two-dimensional reference surface and some assumptions about deformation through the thickness. The REPSIL family of programs is based on theory that restricts them to thin Kirchoff shells with negligible rotary inertia. The coding restricts the programs to shells of uniform thickness with clamped or hinged edges and a plane of symmetry. The RPSL1D version further restricts the program to geometries where the reference surface of the shell is a function of one variable and a symmetry assumption.

Three geometries have been incorporated: axially symmetric shells, slab symmetric shells, and laterally symmetric beams. The standard REPSIL (with some modifications) could be used for these problems, but RPSL1D is much faster and uses about one-fifth as much memory. Further, RPSL1D results should be more accurate; the numerical derivatives with respect to the symmetric variable are replaced by exact derivatives, and spacing for the other variable can be made smaller without unreasonable demands on machine storage or time.

The simpler one-dimensional model makes it easier to experiment with alternate methods of computing. Many changes and additions have been tried. Some of these have been incorporated into RPSL1D, as tabulated in Appendix C, and others discussed in Section 5 and Appendices E and F may be useful in the future.

The purpose of this report is to record RPSL1D, as cataloged in the CDC CYBER system at BRL, to record some user subroutines and useful options, to serve as a user's guide, and to give the basic information for further modifications. This is not a complete guide; the user should also obtain a copy of Reference 2, the user's manual for REPSIL. An understanding of the theory given in Reference 1 is also very helpful. The symbols in the present report, which are sometimes used without sufficient explanation, are those of Reference 1.

Sections 2 and 3 and the list of equations in Appendix A should bridge the gap between REPSIL and RPSL1D. In an effort to keep this report within reasonable bounds, we will not repeat theory given in Reference 1 nor will we repeat much of the information given in the REPSIL user's manual, Reference 2.

---

[1] *J. M. Santiago, "Formulation of the Large Deflection Shell Equations for Use in Finite Difference Structural Response Computer Codes," U.S. Army Ballistic Research Laboratories, Report No. 1571, Feb. 72.*

[2] *J. M. Santiago, H. L. Wisniewski, N. J. Huffington, Jr., "A User's Manual for the REPSIL Code," U.S. Army Ballistic Research Laboratories, Report No. 1744, Oct. 74.*

The RPSL1D user must prepare input as described in Section 6 and must supply proper "user" subroutines as described in Section 4. He may retain the user subroutines with the cataloged program listed in Appendix C, replace them with copies of user subroutines listed in Appendices E and F, or replace them with his own subroutines.

For some particular applications it seems inevitable that some modifications are needed or desirable. A number of possibly useful options are described in Section 5 and listed in Appendices E and F.

The program, as stored in UPDATE form in the CDC 7600, is listed in Appendix C. A complete list of FORTRAN names used in RPSL1D is given in Appendix B along with their definition and some explanation. Two example problems are discussed in Appendix E.

The accessory program for plotting output from RPSL1D is discussed and listed in Appendix D.

The main purpose of this report is to serve as a guide to the use and possible modification of the RPSL1D program. The author's interest in this program is as a programmer. Hence, there is a bias in the presentation toward the frequent use of FORTRAN names and formulation, and the trivial details of programming, at the expense of physical meaning.

2. MODELS

The first one-dimensional version of REPSIL treated axially symmetric shells. This was modified to handle slab symmetric shells, and this in turn changed to treat symmetric beams.

The position, $\underset{\sim}{x}^*$, of any point in the shell may be expressed in terms of a point, $\underset{\sim}{r}$, on the reference surface, the normal, $\underset{\sim}{n}$, to the reference surface at $\underset{\sim}{r}$, and the distance from the reference surface, ζ:

$$\underset{\sim}{x} = \underset{\sim}{r} + \zeta \underset{\sim}{n} \ . \tag{1}$$

Points on the reference surface are expressed in Cartesian coordinates as

$$\underset{\sim}{r} = Y^j \, \underset{\sim}{i}_j \equiv Y^1 \, \underset{\sim}{i}_1 + Y^2 \, \underset{\sim}{i}_2 + Y^3 \, \underset{\sim}{i}_3 \ , \tag{2}$$

where the $\underset{\sim}{i}_j$ are the orthonormal basis for vectors in 3-space. The Y^j, and hence other dependent parameters, are functions of the material (Lagrangian) parameters, ξ^1 and ξ^2, and of time, t.

$$\underset{\sim}{r} = \underset{\sim}{r}(\xi^1, \xi^2, t) = Y^j(\xi^1, \xi^2, t) \, \underset{\sim}{i}_j \ . \tag{3}$$

---

*The symbols of Reference 1 will usually be followed in this report. The tilde below a letter denotes a vector (e.g. $\underset{\sim}{x}$).

Our characterization of axial and slab symmetry is through specialization of the vector $\underset{\sim}{x}$ and the material coordinates ξ^1 and ξ^2. The basic specializations are given in this section. A summary of equations is given in Appendix A.

The type of model is prescribed for RPSL1D by control parameters assigned in the user's subroutine INGEOM.

2.1 Axially Symmetric Shells

An axially symmetric shell is a shell of revolution (see Figure 2.1) such as a circular cylinder or the surface of a truncated right circular cone. A shell of revolution is the surface generated by revolving a plane curve (actually two parallel curves so the shell has thickness) about a line, called the axis of revolution, in its plane. Every point of the revolving curve describes a circle whose center is on the axis. For our shell, the curve must be smooth and must not intersect the axis in the region of interest.

We study the motion of the shell by following the motion of the curve formed by the intersection of the shell's reference surface and a plane through the axis. We do not allow twisting in our model, so the initial motion, and any loading, must be in this plane. Because of the axial symmetry, the internal forces are all in this plane.

Let $\xi^1 = \theta$, $\xi^2 = \xi$, and introduce the orthonormal base vectors $(\hat{\underset{\sim}{r}}, \hat{\underset{\sim}{\theta}}, \hat{\underset{\sim}{k}})$ which vary with angle θ. These are related to $(\underset{\sim}{i}_1, \underset{\sim}{i}_2, \underset{\sim}{i}_3)$ by the relations

$$\hat{\underset{\sim}{r}} \equiv \hat{\underset{\sim}{r}}(\theta) = \cos\theta\, \underset{\sim}{i}_3 + \sin\theta\, \underset{\sim}{i}_1 , \tag{4}$$

$$\hat{\underset{\sim}{\theta}} \equiv \hat{\underset{\sim}{\theta}}(\theta) = -\sin\theta\, \underset{\sim}{i}_3 + \cos\theta\, \underset{\sim}{i}_1 , \tag{5}$$

$$\hat{\underset{\sim}{k}} \equiv \underset{\sim}{i}_2 . \tag{6}$$

Then,

$$\underset{\sim}{x} \equiv \underset{\sim}{x}(\theta, \xi, t) = R\, \hat{\underset{\sim}{r}} + Z\, \hat{\underset{\sim}{k}} \tag{7}$$

where

$$R \equiv R(\xi, t) = \{(Y^3)^2 + (Y^1)^2\}^{\frac{1}{2}} , \tag{8}$$

$$Z \equiv Z(\xi, t) = Y^2 . \tag{9}$$

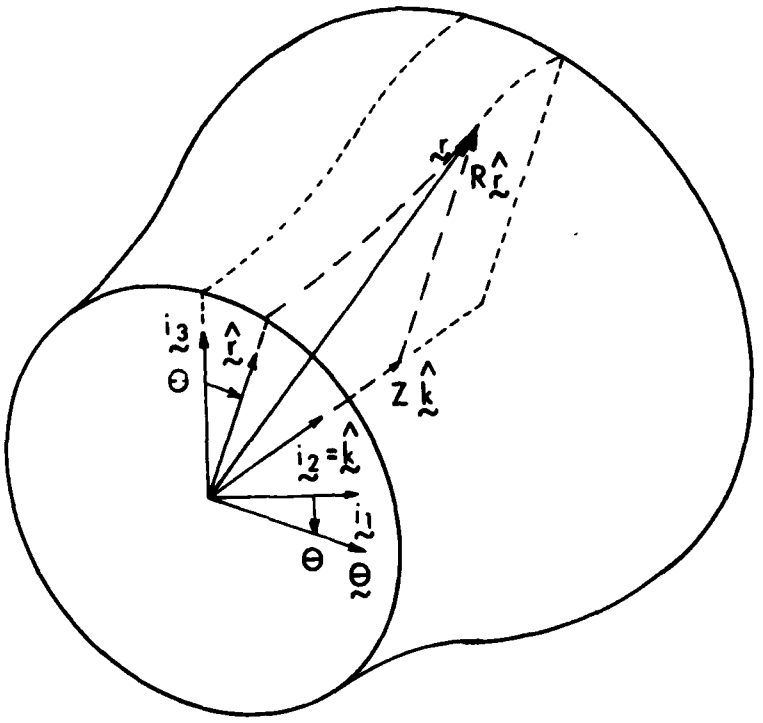


Figure 2.1 - Model for axial symmetry showing the two orthonormal bases $(\underset{\sim}{i}_1, \underset{\sim}{i}_2, \underset{\sim}{i}_3)$ and $(\hat{\underset{\sim}{r}}, \hat{\underset{\sim}{\theta}}, \hat{\underset{\sim}{k}})$ with origin at the center of end 1 of the shell, and a vector $\underset{\sim}{r}$ to the surface of the axisymmetric shell.

The same formulation holds for every angle θ. Since evaluation is needed at only one angle, we naturally choose $\theta = 0$. Then,

$$\underset{\sim}{r}(0,\xi,t) = R(\xi,t)\ \hat{\underset{\sim}{r}}(0) + Z(\xi,t)\ \hat{\underset{\sim}{k}}$$

$$\equiv Y^3(0,\xi,t)\ \underset{\sim}{i}_3 + Y^2(0,\xi,t)\ \underset{\sim}{i}_2 \ . \qquad (10)$$

To test this new program for axial symmetry, the same problem (an initially cylindrical shell subjected to a uniform impulsive load) was simulated with both programs. Both programs were run undamped for 200 time cycles and then damped. Both self terminated on time cycle 262. The results were very close. The maximum difference noted in displacement was less than $2*10^{-8}$. Since the displacement was about unity, $2*10^{-8}$ was also the relative difference in displacement. Strains and displacement increments are very sensitive to program differences. The largest relative difference noticed in either was 10^{-5}. (There were relative differences of about 0.003 in all the printed energy values. This was because the circumference of the right circular cylinder was a circle in the axial symmetric program but was a 48-sided regular polygon in the standard REPSIL.)

2.2 Slab Symmetric Shells

Slab symmetric shells may be characterized by the displacement of any point being a function of its initial distance, ξ^2, from end 1 (see Figure 2.2).

Let $Y^1 = \xi^1$, $Y^2 = Y^2(\xi^2,t)$, and $Y^3 = Y^3(\xi^2,t)$. Then,

$$\underset{\sim}{r}(\xi^1,\xi^2,t) = \xi^1 \underset{\sim}{i}_1 + Y^2(\xi^2,t)\ \underset{\sim}{i}_2 + Y^3(\xi^2,t)\ \underset{\sim}{i}_3 \ . \tag{11}$$

To utilize the formulation and programming for axial symmetry, we renamed variables

$$\xi^2 = \xi, \quad Y^3 = R, \quad \text{and } Y^2 = Z \ , \tag{12}$$

and confined evaluation to the plane $Y^1 = 0$. Then,

$$\begin{aligned} \underset{\sim}{x} \equiv \underset{\sim}{x}(0,\xi,t) &= R(\xi,t)\ \underset{\sim}{i}_3 + Z(\xi,t)\ \underset{\sim}{i}_2 \\ &\equiv Y^3(\xi,t)\ \underset{\sim}{i}_3 + Y^2(\xi,t)\ \underset{\sim}{i}_2 \ . \end{aligned} \tag{13}$$

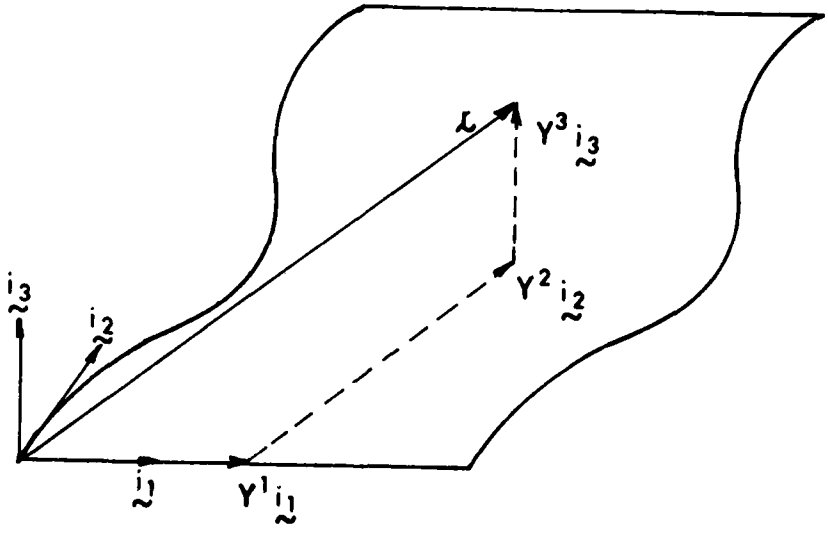


Figure 2.2 - Model for Slab Symmetry.

To maintain slab symmetry, the loading must not contain any component in the $\underset{\sim}{i}_1$ direction.

For both axial and slab symmetry, Z is a distance along the $\underset{\sim}{i}_2$ axis. For axial symmetry, R is the radial distance from the axis of rotation. For slab symmetry, R is the perpendicular distance from the $Y^3 = 0$ plane.

The initial slab symmetric coding was also tested by comparing with results from a standard REPSIL run. (A minor coding change was needed to insert cyclic boundary conditions in REPSIL.) The results were nearly identical. The maximum relative errors in any variables were of the order of 10^{-13}.

2.3 Laterally Symmetric Beams

One principal difference between slab symmetric shells and beams is in the change of concept from area to unit length (e.g. pressure, i.e. force/unit area, for shells and force/unit length for beams.) The original beam considered was a narrow rectangular beam. It was treated as a special case of slab symmetry with a uniaxial stress-strain relation replacing the biaxial one. The uniaxial stress-strain was imposed by using a new subroutine BMSTRS, instead of subroutine STRESS, for beams. By setting the width, DETA1 ($\Delta\xi^1$), to unity, the concept of surface area became length without any further changes.

Treating beams that were not rectangular was more involved. It required changes in fact as well as in concept. The main change was for integrals through the thickness. The simplest form of numerical integration through the thickness is used in the standard REPSIL:

$$\int_{-h/2}^{+h/2} f(\zeta)\, d\zeta \sim \Delta\zeta \sum_{k=1}^{L} f(\zeta_k) \ . \tag{14}$$

The thickness, h, is divided into L layers of equal thickness, $\Delta\zeta = h/L$, and the function to be integrated is evaluated at each ζ_k, where ζ_k is in the center of the k'th layer. The parameter ζ is measured from the reference surface in the center of the shell.

For a beam, an integral through the thickness is replaced by an integral over the cross-sectional area. When multiplied by the proper metric, this becomes the quantity per unit surface area. For a beam, the desired result is the quantity per unit length. For a rectangular beam we could simply multiply the previous result by the width of the beam, $\Delta\xi^1$:

$$\int_A f(\zeta)\, dA = \Delta\xi^1 \int_{-h/2}^{+h/2} f(\zeta)\, d\zeta \sim \Delta\xi^1\, \Delta\zeta \sum_{k=1}^{L} f(\zeta_k) \ . \tag{15}$$

This has been replaced by a more general form for laterally symmetric but not necessarily rectangular beams:

$$\int_A f(\zeta)\, dA \sim \sum_{k=1}^{L} W_k\, f(\zeta_k) \qquad (16)$$

where the ζ_k are not necessarily evenly spaced and W_k is the area associated with ζ_k. The ζ_k and W_k should be assigned so that $\sum W_k$ is the cross-sectional area of the beam, $\sum \zeta_k W_k = 0$ (so the reference surface is the neutral surface for bending), and $\sum (\zeta_k)^2 W_k$ is the area moment of inertia of the cross section. (This form of integral approximation is also suitable for Gaussian integration, which is preferred for shells; see Section 3.4.)

See the discussion of INGEOM for a beam (Section 4.1.1) for a particular example. The user subroutine INGEOM for beams must also supply the upper limit of the beam, ZU $\equiv \zeta_u$. Subroutine START assigns $\zeta_\ell \equiv$ ZL = ZU - h. For beams, ZU and ZL merely locate positions, upper and lower respectively, at which surface strains are computed for output. The shell versions set ZU = h/2, ZL = -h/2, and ZU is used in START to compute a number of program constants.

Results from RPSL1D for small elastic displacements for vibrations initiated in the fundamental transverse modes for clamped, hinged, and cantilever beams are in good agreement with simple beam theory. An example of buckling from lateral forces on the ends also agreed with theory.

3. CHANGES FROM REPSIL

The principal reason for programming a one-dimensional version of REPSIL was to have a simple vehicle for testing possible changes before implementing them in REPSIL. However, the program proved very useful for a variety of problems. Many changes have been coded. Some of these were purely experimental, but most of them were for particular problems. Most of the changes have been superseded or were not considered general enough to transfer to the CDC computer. Some of the changes have been incorporated as part of RPSL1D, as listed in Appendix C, and some are retained as options.

In this section we will outline the changes made to REPSIL to create the RPSL1D program stored in the CDC 7600 as of September 1978, and describe the new boundary condition implementation, the new differencing scheme, Gaussian integration, and the new stability criteria.

3.1 Outline of the Development of RPSL1D

Initially, the REPSIL program, listed in Reference 2, was changed to a one-dimensional code for axially symmetric shells (see Section 2.1). A change of notation was introduced to eliminate errors.

This program was modified to include slab symmetry (Section 2.2).

This was then modified to include beams with a vertical axis of symmetry (Section 2.3).

The method of imposing boundary conditions was changed by inserting artificial, external points as described in Section 3.3, and the divided difference approximations for partial derivatives were changed (see Section 3.2) to a form using only central differences. The system was made more compact by evaluating at mid-mesh points.

The code was changed to permit symmetry at either end and permit a free end at end 2. This change was intended for beams only. (An option called APLFRC, see Section 5.4, which permits forces to be applied at either or both ends of beams is available.)

Coding was added to monitor maximum deflection and extreme surface strains, and to print these extremes along with the surface strain prints.

The computation of surface strains was revised. The original surface strain computation was a copy of that described on pages 50 through 53 of Reference 2 except that a_{12} and b_{12} were zero and ζ_u and ζ_ℓ replace h/2 and -h/2. This was changed so that the interpolation in the non-zero covariant components of strain, ε_{11} and ε_{22} ($\varepsilon_{\alpha\beta} = \frac{1}{2}(\Sigma\Delta a_{\alpha\beta} \pm h\ \Sigma\Delta b_{\alpha\beta})$) was replaced by interpolation in the changes in the covariant components of the middle surface metric, $\Sigma\Delta a_{11}$ and $\Sigma\Delta a_{22}$, and of the second fundamental tensor, $\Sigma\Delta b_{11}$ and $\Sigma\Delta b_{22}$. Component $\Sigma\Delta a_{22}$ is computed between mesh points.

Optional Gaussian integration through the thickness of shells was introduced.

Several minor, mostly cosmetic, changes were made.

The coding was transferred from the BRLESC computer to the CDC 7600 (with more minor changes) and stored in the UPDATE form.

3.2 New Difference Operators

The primary purpose of the REPSIL family of programs is to solve the equations of motion. This is done by a sequence of equations (see Reference 1, pp. 99-106) that carry the solution forward one time step.

Difference approximations of derivatives with respect to the material coordinates, ξ^1 and ξ^2, are required at two points in this sequence of equations. First, we need approximations to the first and second derivatives of position, $\underset{\sim}{x}$, and the incremental displacement, $\Delta\underset{\sim}{x}$, to define the local geometry. Finally, we need approximations for first partial derivatives of the components of the stress resultant tensor, $\hat{\underset{\sim}{N}}^{*\alpha}$, and second partial derivatives of the normal vector components of the bending resultant tensor, $\hat{\underset{\sim}{M}}^{*\alpha\beta}$, for the equations of motion.

The REPSIL equations of motion for a mesh point, as given in Reference 1, are

$$\Delta\underset{\sim}{u}^{+} = \Delta\underset{\sim}{u} + \frac{(\Delta t)^2}{\gamma^*_0}\left[\frac{\partial^2\hat{\underset{\sim}{M}}^{*\alpha\beta}}{\partial\xi^\alpha\partial\zeta^\beta} + \frac{\partial\hat{\underset{\sim}{N}}^{*\alpha}}{\partial\xi^\alpha} + \underset{\sim}{P}^*\right] . \tag{17}$$

The change in the displacement of the point from time t to t + Δt is $\Delta\underset{\sim}{u}^{+}$. The position at time t + Δt is $\underset{\sim}{x}^{+} = \underset{\sim}{x} + \Delta\underset{\sim}{u}^{+}$, where $\underset{\sim}{x}$ is the position at time t. The mass per unit initial middle surface area, γ^*_0, is constant for a given point. (The term in brackets has units of force per initial unit area.) $\underset{\sim}{P}^* = -P\, a^{\frac{1}{2}}\, \underset{\sim}{n}$ is the force per initial unit area along the normal, $\underset{\sim}{n}$, due to a pressure P. (a is the determinant of the covariant metric tensor.) The one-dimensional assumptions remove a number of the terms in the equations of motion. There are no terms in the $\underset{\sim}{i}_1$ direction, $\hat{\underset{\sim}{M}}^{*12} = \hat{\underset{\sim}{M}}^{*21} = 0$, and the derivatives with respect to ξ^1 are simple functions which are computed explicitly. The equations of motion become

$$\Delta\underset{\sim}{x}^{+} = \Delta\underset{\sim}{x} + \frac{(\Delta t)^2}{\gamma^*_0}\left[\frac{\partial^2\hat{\underset{\sim}{M}}^{*22}}{(\partial\xi^2)^2} + \frac{\partial\hat{\underset{\sim}{N}}^{*2}}{\partial\xi^2} + \underset{\sim}{P}^* - (\hat{\underset{\sim}{M}}^{*11}\cdot\underset{\sim}{i}_2 + \hat{\underset{\sim}{N}}^{*1}\cdot\underset{\sim}{i}_1)\,\underset{\sim}{i}_2\right], \tag{18}$$

$$\underset{\sim}{x}^{+} = \underset{\sim}{x} + \Delta\underset{\sim}{x}^{+} . \tag{19}$$

(The transformation to this equation is outlined in Appendix A.) The vectors $\hat{\underset{\sim}{M}}^{*22}$, $\hat{\underset{\sim}{N}}^{*2}$, and their derivatives and $\underset{\sim}{P}^*$ and $\hat{\underset{\sim}{M}}^{*11}$ may all have components in both the $\underset{\sim}{i}_3$ and $\underset{\sim}{i}_2$ directions. $\hat{\underset{\sim}{N}}^{*1}$ has a component in the $\underset{\sim}{i}_1$ direction only.

For slab symmetry and beams, the term $(\hat{\underset{\sim}{M}}^{*11}\cdot\underset{\sim}{i}_2 + \hat{\underset{\sim}{N}}^{*1}\cdot\underset{\sim}{i}_1)$ is deleted. For beams, γ^*_0 is mass/unit length and the term in brackets is force/unit length.

The difference scheme used in the standard REPSIL has two defects which have occasionally been troublesome: the use of forward, or backward, differences at fixed edges may introduce a slowly developing instability which cannot be eliminated by a reduction in the time step. (The presence of this instability could be detected by an eigenvalue analysis of the linearized equations of motion. We have carried out this analysis for a few cases with very small grids. Unfortunately, for most useful meshes, the eigenvalue analysis would take longer than the REPSIL program.) The other defect is that the differences for first derivatives are not compact enough. (As an experiment with the initial RPSL1D code, alternate points in a slab symmetric run were at rest or given an initial axial velocity. The points initially at rest remained at rest. The motion of the moving points produced a reasonable solution.) These two defects have been removed from RPSL1D (and from a modified version of REPSIL) by modeling fixed boundaries with artificial external points and by introducing new difference operators which compute and use forces between mesh points as well as at mesh points.

Three sets of central difference operators are used in RPSL1D to compute first and second differences with respect to the Lagrangian coordinate $\xi = \xi^2$. To simplify the notation, replace ξ^2 by y and use the notation $f_y(n)$ and $f_{yy}(n)$ to represent the approximations to $\partial f/\partial \xi^2$ and $\partial^2 f/(\partial \xi^2)^2$, respectively, evaluated at the n'th mesh point.

The first set of difference operators is used in subroutine GRAD to approximate derivatives of position components, and derivatives of components of displacement increments, at mesh points:

$$f_y(n) = \{f(n+1) - f(n-1)\}/(2\,\Delta y) , \tag{20}$$

$$f_{yy}(n) = \{f(n+1) - 2\,f(n) + f(n-1)\}/(\Delta y)^2 . \tag{21}$$

The second set of difference operators is used in subroutine GRAD to approximate the same derivatives at mesh midpoints:

$$f_y(n+\tfrac{1}{2}) = \{f(n+1) - f(n)\}/\Delta y , \tag{22}$$

$$f_{yy}(n+\tfrac{1}{2}) = \tfrac{1}{2}\{f(n+2) - f(n+1) - f(n) + f(n-1)\}/(\Delta y)^2 . \tag{23}$$

The third set of central difference operators is used in subroutine MOTION to compute terms for the equations of motion:

$$f_y(n) = \{f(n+\tfrac{1}{2}) - f(n-\tfrac{1}{2})\}/\Delta y , \tag{24}$$

$$f_{yy}(n) = \{f(n+1) = 2\,f(n) + f(n-1)\}/(\Delta y)^2 . \tag{25}$$

These difference operators, which are used at fixed edges as well as at internal points, are all central difference operators. The resulting system of difference equations is stable with the proper choice of time step, Δt. Unfortunately, when the program is run for a beam with a free end at end 2, say n = N, backward difference operators have to be used at the free end. In GRAD, for a free end, we use:

$$f_y(N) = \{3\, f(N) - 4\, f(N-1) + f(N-2)\}/(2\,\Delta y)\ , \tag{26}$$

$$f_{yy}(N) = \{2\, f(N) - 5\, f(N-1) + 4\, f(N-2) - f(N-3)\}/(\Delta y)^2\ , \tag{27}$$

$$f_y(N-\tfrac{1}{2}) = \{f(N) - f(N-1)\}/\Delta y\ , \quad \text{(as before)} \tag{28}$$

$$f_{yy}(N-\tfrac{1}{2}) = \tfrac{1}{2}\{3\, f(N) - 7\, f(N-1) + 5\, f(N-2) - f(N-3)\}/(\Delta y)^2\ . \tag{29}$$

In MOTION, for a free end, we have:

$$f_y(N) = \{f(N) - f(N-\tfrac{1}{2})\}/(\Delta y/2)\ , \tag{30}$$

$$f_{yy}(N) = \{2\, f(N) - 5\, f(N-1) + 4\, f(N-2) - f(N-3)\}/(\Delta y)^2\ . \tag{31}$$

It is also necessary to approximate $\hat{\underset{\sim}{N}}^{*2} = -\partial\hat{\underset{\sim}{M}}^{*22}/\partial\xi$ at the free end. This is done in subroutine RESULT with

$$\hat{\underset{\sim}{N}}^{*2}(N) = \tfrac{1}{2}\{4\ \hat{\underset{\sim}{M}}^{*22}(N-1) - \hat{\underset{\sim}{M}}^{*22}(N-2)\}/\Delta\xi\ , \tag{32}$$

because

$$\hat{\underset{\sim}{M}}^{*22}(N) = 0.0\ . \tag{33}$$

3.3 External Points and End Conditions

The end conditions for a fixed or symmetric end is imposed in RFSL1D by the initial positioning and the incrementing of a fictional external point. The following notation is used to explain this:

- $\underset{\sim}{r}$ A vector to a point on the reference surface at time t. ($\underset{\sim}{r} = R\,\underset{\sim}{i}_3 + Z\,\underset{\sim}{i}_2$)
- $\underset{\sim}{r}^+$ A vector to a point on the reference surface at time t + Δt.
- $\Delta\underset{\sim}{r}$ $\underset{\sim}{r}^+ - \underset{\sim}{r}$
- $\underset{\sim}{r}_B$ A vector to the end point.
- $\underset{\sim}{r}_I$ A vector to the internal mesh point adjacent to $\underset{\sim}{r}_B$.
- $\underset{\sim}{r}_E$ A vector to the external point adjacent to $\underset{\sim}{r}_B$.
- $\underset{\sim}{n}_B$ The unit normal to the reference surface at $\underset{\sim}{r}_B$. ($\underset{\sim}{n}_B = n_r\,\underset{\sim}{i}_3 + n_k\,\underset{\sim}{i}_2$)
- $\underset{\sim}{\tau}$ The unit tangent to the reference surface at $\underset{\sim}{r}_B$. ($\underset{\sim}{\tau} = -n_k\,\underset{\sim}{i}_3 + n_r\,\underset{\sim}{i}_2$)

The initial position of external points is established in subroutine BOUNDR and the corresponding increments are inserted in BOUNDU. The initial normal at clamped end points is supplied in a new user subroutine INNORM.

3.3.1 Hinged Ends. The initial position of an artificial external point at a hinged end is

$$\underset{\sim}{r}_E = 2\,\underset{\sim}{r}_B - \underset{\sim}{r}_I \,. \tag{34}$$

The increment is

$$\Delta\underset{\sim}{r}_E = -\Delta\underset{\sim}{r}_I \,. \tag{35}$$

We also set

$$\Delta\underset{\sim}{r}_B = 0 \,. \tag{36}$$

These inserts impose $\underset{\sim}{r}_B$ is constant and $\hat{\underset{\sim}{M}}^{*22} = 0$ at the boundary. These are the conditions for a hinged end as given in Reference 1.

3.3.2 Clamped Ends. The initial position of an artificial external point adjacent to a clamped end is

$$\underset{\sim}{r}_E = \underset{\sim}{r}_I - 2\{(\underset{\sim}{r}_I - \underset{\sim}{r}_B) \cdot \underset{\sim}{\tau}\}\,\underset{\sim}{\tau} \,. \tag{37}$$

The corresponding increment is

$$\Delta\underset{\sim}{r}_E = \Delta\underset{\sim}{r}_I - 2\{\Delta\underset{\sim}{r}_I \cdot \underset{\sim}{\tau}\}\,\underset{\sim}{\tau} \,. \tag{38}$$

We also impose

$$\Delta\underset{\sim}{r}_B = 0 \,. \tag{39}$$

These inserts force $\underset{\sim}{r}_B$ and $\underset{\sim}{n}_B$ to remain constant. These are the conditions for a clamped end given in Reference 1. (The components of $\underset{\sim}{n}_B$ are set in a user subroutine INNORM and never changed for a clamped end.)

3.3.3 Symmetric Ends. The treatment of symmetric ends is simply a reduction of that for symmetric edges in REPSIL. A symmetry plane is assumed perpendicular to the basis vector $\underset{\sim}{i}_2$. The initial position of an external point adjacent to a symmetry end is

$$\underset{\sim}{r}_E = \underset{\sim}{r}_I - 2\{(\underset{\sim}{r}_I - \underset{\sim}{r}_B) \cdot \underset{\sim}{i}_2\}\,\underset{\sim}{i}_2 \,. \tag{40}$$

The change in $\underset{\sim}{r}_E$ at each time step is

$$\Delta\underset{\sim}{r}_E = \Delta\underset{\sim}{r}_I - 2\{\Delta\underset{\sim}{r}_I \cdot \underset{\sim}{i}_2\}\,\underset{\sim}{i}_2 \,. \tag{41}$$

These are the same equations used for clamped ends with $\underset{\sim}{i}_2 = \underset{\sim}{\tau}$.

The external points at symmetry ends represent real points. They are not artificial like those outside the fixed ends. Just as in the standard REPSIL, we need to know membrane and bending resultant components ($\hat{\underset{\sim}{N}}^{*2}$ and $\hat{\underset{\sim}{M}}^{*22}$, respectively) outside the symmetry end. We have

$$(\hat{\underset{\sim}{M}}^{*22})_I \equiv \hat{M}^{*22}\, \underset{\sim}{n}_I = (FM_r^{22})_I\, \underset{\sim}{i}_3 + (FM_k^{22})_I\, \underset{\sim}{i}_2 , \tag{42}$$

defined at $\underset{\sim}{r}_I$, and

$$(\hat{\underset{\sim}{N}}^{*1})_I = (FN_\theta^1)_I\, \underset{\sim}{i}_1 , \tag{43}$$

$$(\hat{\underset{\sim}{N}}^{*2})_I = (FN_r^2)_I\, \underset{\sim}{i}_3 + (FN_k^2)_I\, \underset{\sim}{i}_2 , \tag{44}$$

defined between $\underset{\sim}{r}_I$ and $\underset{\sim}{r}_B$. The external values needed are

$$(FM_r^{22})_E = (FM_r^{22})_I , \tag{45}$$

$$(FM_k^{22})_E = -(FM_k^{22})_I , \tag{46}$$

$$(FN_\theta^1)_E = (FN_\theta^1)_I , \tag{47}$$

$$(FN_r^2)_E = -(FN_r^2)_I , \tag{48}$$

$$(FN_k^2)_E = (FN_k^2)_I . \tag{49}$$

3.3.4 Free Ends. One of the first problems posed for the RPSL1D code was for a cantilever I-beam. To model this beam a free end was needed. The conditions for a free end (see Ref. 2, Eq. 6.52) are imposed by

$$\hat{\underset{\sim}{M}}^{*22} = 0.0 , \quad \hat{\underset{\sim}{N}}^{*2} = -\partial \hat{\underset{\sim}{M}}^{*22}/\Delta\xi . \tag{50}$$

Regrettably, it is necessary to use non-central differences to approximate this derivative, and other derivatives, at free ends (see Section 3.2).

The cataloged program permits end 2 to be free but not end 1. (The optional coding called APLFRC, see Section 5.4, allows either end to be free with, or without, applied forces. The applied forces, or absence of forces, are supplied by a user's subroutine called ENDFRC.)

3.4 Optional Gaussian Integration

The original RPSL1D programming approximated integration through the thickness of shells by Riemann sums assuming equal layers about evenly spaced points. This is replaced with Gaussian integration only if IGAUSS is 1 and IB is not 1. (These control parameters are set in the user subroutine INGEOM, see Section 4.1.) IB = 1 specifies a beam. We do not use Gaussian integration with beams. We use Riemann sums with unevenly spaced points and "weights" equal to their associated cross-sectional area (as explained in Section 2.3). For the Riemann sums with L equal layers we choose

$$\zeta_k = \tfrac{1}{2}\, h\, (1 - (2\,k - 1) / L)\ ,\ (k = 1,2,\cdots,L)\ , \tag{51}$$

$$\Delta\zeta = h/L\ , \tag{52}$$

$$\zeta_u = h/2\ , \tag{53}$$

$$\zeta_\ell = -\ h/2\ , \tag{54}$$

and approximate integration through the thickness by

$$\int_{\zeta_\ell}^{\zeta_u} f(\zeta)\ d\zeta \sim \sum_{k=1}^{L} f(\zeta_k)\ \Delta\zeta\ , \tag{55}$$

where h is the shell thickness, ζ_u and ζ_ℓ are the distances from the reference surface (the middle surface) to the upper and lower surfaces, respectively. L is the number of equally spaced layers, and ζ_k is the center of the k'th layer.

For Gaussian integration, we replace the ζ_k by

$$\zeta_k = \tfrac{1}{2}\, h\, x_{kL}\ , \tag{56}$$

set

$$W_k = W_{kL}\ , \tag{57}$$

and use the approximation

$$\int_{\zeta_\ell}^{\zeta_u} f(\zeta)\ d\zeta \sim (h/2) \sum_{k=1}^{K} f(\zeta_k)\ W_k\ . \tag{58}$$

where the x_{kL} and W_{kL} are from tables for Gaussian integration (e.g. Table 25.4, Handbook of Mathematical Functions, National Bureau of Standards). The x_{kL} are zeros of the Legendre Polynomial

$$P_L(x) = \frac{1}{2^L\ L!}\ \frac{d^L\ (x^2 - 1)^L}{(dx)^L} \tag{59}$$

Two sets of integrals are affected: the integral for strain energy in subroutine STRESS, and the integrals for force and moment resultants in subroutine RESULT.

The Riemann sums for the force and moment resultants at a point N are of the form

$$I_N = TA \sum_k f_N(\zeta_k)\ , \tag{60}$$

where

$$TA = \Delta\zeta = h/L \ . \tag{61}$$

The $f_N(\zeta_k)$ signifies the value of the appropriate function at mesh point N and integration station ζ_k, where the ζ_k are in the center of evenly spaced layers.

All that is needed to change these sums to Gaussian integration is to select the proper ζ_k and W_k, replace $f_N(\zeta_k)$ by $W_k\ f_N(\zeta_k)$, and set TA to h/2. For example,

$$I_N = (h/2) \sum_{k=1}^{K} W_k\ f_N(\zeta_k) \ . \tag{62}$$

(The form is the same for approximations of integrals over cross-sectional area for beams. This is done with Riemann sums but with unevenly spaced ζ_k. The ζ_k, and W_k equal to the area associated with ζ_k, are selected in the user subroutine INGEOM, and TA is set to 1.0 in START.)

3.5 Convergence and Stability

Some examination of convergence and stability is essential for any numerical solution of differential equations. The usual method of examining convergence is to find solutions for a particular problem using an increasing number of meshes until these solutions converge. Such tests with the REPSIL family of programs have shown remarkable insensitivity to the number of meshes for problems with large plastic deformation. In those cases tested it seems that the maximum deflection, usually at the center, decreases slightly with additional meshes, and the strain and deflection increase slightly near the edges. For elastic vibration problems, more meshes may be needed to permit high vibration modes. The choice of mesh, and hence the accuracy, must ultimately be decided by the user.

Numerical solutions of differential equations by explicit methods such as that used with the REPSIL programs will be unstable, because any introduced error grows, unless the increment in the independent variable, Δt in our case, is small enough. Considerable time and effort have been expended in studying the stability for the various REPSIL codes. Even a cursory discussion of the stability analysis is beyond the scope of this report. The following bounds, based on a linear analysis of the non-linear difference equations of motion for RPSL1D assuming the central difference operators, are used in RPSL1D:

$$\Delta t_M = \Delta\xi/C \ , \tag{63}$$

$$\Delta t_B = \tfrac{1}{2}(\Delta\xi)^2/(CC_2) \ . \tag{64}$$

Here, Δt_M and Δt_B are the stable time step for the linearized membrane and bending equations of motion, respectively. C is the longitudinal wave speed.

$$C = (E/\rho)^{\frac{1}{2}} \text{ , for beams.} \tag{65}$$

$$C = \{E/\ \rho(1 - \nu^2)\}^{\frac{1}{2}} \text{ , for shells.} \tag{66}$$

Here, E is Young's modulus of Elasticity, ρ is mass density, and ν is Poisson's ratio. C_2 is the approximation of $\{ \int_A \zeta^2 \, dA \, / \int_A dA \}^{\frac{1}{2}}$.

For beams,

$$C_2 = \{ \sum W_k \, (\zeta_k)^2 / \sum W_k \}^{\frac{1}{2}} . \tag{67}$$

For shells h units thick, using Riemann sums with L equal layers,

$$C_2 = \{(1 - 1/L^2)\ h^2/12\}^{\frac{1}{2}} . \tag{68}$$

For shells with Gaussian integration the approximation is exact:

$$C_2 = (h^2/12)^{\frac{1}{2}} . \tag{69}$$

The program chooses a two digit truncated value 0.95 times the minimum of Δt_M and Δt_B. This is used instead of the input Δt on input card 3 if it is smaller than the input value given, or if the input value is negative or zero.

The term $\Delta\xi$ used by RPSL1D in these equations is the DETA2 supplied by the user through subroutine INGEOM. The equations produce critical values for Δt_M and Δt_B if $\Delta\xi$ is the minimum distance between mesh points. Both Δt_M and Δt_B have been shown to be close to the limits for stability for runs with fixed ends and a linearly elastic stress-strain relation. The user should examine all results for numerical instability. This shows up first as rapid vibrations in strain plots. The inclusion of plasticity tends to damp these vibrations, but they can still be detected. The user should be particularly critical if there is a free or forced end (non-central differences are used), or if the structure is compressed. The stability formuli are based on the assumption that $\Delta\xi$ is the minimum distance between mesh points. The factor 0.95 permits some compression, but only up to 5% compression if Δt_M is critical, or 2.5% if Δt_B is critical. We have not been able to prove the stability of the RPSL1D solutions when non-central difference equations are used, but we have not detected instability when a time step that satisfies the stability criteria is used.

4. USER SUBROUTINES

There are four, or possibly five, subroutines in RPSL1D which the user must either write, or select from existing versions. These "user" subroutines describe the geometry and loading. Three of them, INGEOM, PRESS, and INVEL, are one-dimensional versions of subroutines described in Reference 2. Subroutine INNORM was added to supply the unit normal to the reference curve at clamped ends. Subroutine ENDFRC must be included, to supply the forces and moments on the ends of a beam, if the optional coding APLFRC, which assumes this type of loading (see Section 5.4), is added to the RPSL1D program. Some of these subroutines are always called and some are called under control of input parameters. Any of the user supplied subprograms may require input (this is under the control of the user). The input for them follows all the other input (input is discussed in Section 6). The user subroutines are listed below in the order they are called, with the conditions for calling.

INGEOM is called from START shortly after the start of each run.

INNORM is called immediately after INGEOM from START.

PRESS is called from the initiation part of the main program, RPSL1D, if LOAD $\neq 0$. It will be called each time cycle thereafter as long as LPRESS $\geq$ NCYCLE.

INVEL is called in the initiation sequence in RPSL1D if LOAD ≤ 0.

ENDFRC is called from RESULT each cycle, except cycle 0, if the APLFRC optional coding is included in the program.

Information is brought to each of the user subroutines, and carried back to the rest of the program, through COMMON. The RPSL1D program is intended to be stored in the UPDATE format. All the COMMON is contained in a COMDECK called MAIN which is listed at the beginning of Appendix C. This COMDECK is inserted into a subroutine with the directive *CALL MAIN. Of course, information can also be passed from one user subroutine to another through labeled COMMON. The parameters in the following list are commonly needed in the user subroutines.

NMESH The number of mesh intervals in the length.

LAYER The number of layers in a cross section.

N1B = 2 The index of the mesh point at end 1.

N2B = N1B + NMESH The index of the mesh point at end 2.

N1V and N2V The range of indices of mesh points that move.

4.1 INGEOM

Subroutine INGEOM is called from START. It supplies the initial geometry and some control parameters. The parameters supplied are listed below.

IB This is the dominant control parameter. It should be set to 1 or 0. if IB > 0, a beam is assumed. If IB $\leq$ 0, it is not.

RADIUS This parameter separates radial symmetry from slab symmetry. If RADIUS > 0, radial symmetry is assumed and the parameter IS is set to 1 in START. Otherwise, IS is set to 2 in START. (RADIUS is superseded by IB. If IB > 0, RADIUS is set to 0 in START.)

DETA1 ($\Delta\xi^1$) This parameter is a multiplier for energy computations. It should be the width of the reference surface. (If IB = 1, DETA1 is set to 1.0 in START. This makes the concept of surface area and beam length interchangeable.)

DETA2 ($\Delta\xi$) This is the distance between mesh points in the material coordinate ξ (ξ^2 of Ref. 1, and η^2 of Ref. 2). (The stability criteria assume that DETA2 is the minimum distance between mesh points, and for some other coding it is tacitly assumed that ξ is arc length from end 1.)

R(N), Z(N) (N1B $\leq$ N $\leq$ N2B) The position vector of the N'th mesh point on the reference curve is $\underset{\sim}{r}_N$ = R(N) $\underset{\sim}{i}_3$ + Z(N) $\underset{\sim}{i}_2$. The point $\underset{\sim}{r}_N$ corresponds to the material coordinate point ξ = (N - 1) DETA2.

ETAD2 and ETAG2(I), I=1,NSTRN The previously read parameters ETAD2 and ETAG2(I) may need to be converted to the same units as DETA2.

IGAUSS Set IGAUSS = 1 for Gaussian integration (see Section 3.4).

Extra Parameters for Beams

If IB = 1, the cross section of the beam must be described. It is defined as follows:

ZU (ζ_u) The distance from the reference surface to the upper surface of the beam.

ZETA(K), K=1,LAYER (ζ_k) These are the positions of integration stations through the thickness relative to the reference surface, the neutral axis for bending of the beam. (ZETA(1) < 0, ZETA(LAYER) > 0)

W(K), K=1,LAYER (W_k) The cross-sectional area associated with ZETA(K).

It is assumed that

$$\sum_k W_k = \text{Area of the beam's cross section.}$$

$$\sum_k (\zeta_k)^2 W_k = \text{Moment of inertia.}$$

$$\sum_k \zeta_k W_k = 0 \quad \text{(i.e. } \zeta = \text{zero on the neutral surface)}.$$

4.1.1 INGEOM for a Flat Plate or Beam. This is the INGEOM stored with RPSL1D (see Appendix C).

If the run is for slab symmetric motion of an initially flat plate, there is only one input card. The input is simply length, SLABL, and one-half the width, SLABW, as it is for a similar standard REPSIL subroutine. First, the subroutine reads SLABL, SLABW, and IB with FORMAT (2E10.4,I5). The program sets

$$\begin{aligned} &\text{RADIUS} = 0.0 \ , \\ &\text{DETA1} = 2.0 * \text{SLABW} \ , \\ &\text{DETA2} = \text{SLABL/FLOAT(NMESH)} \ , \\ &\left.\begin{aligned} &\text{R(N)} = 0.0 \\ &\text{Z(N)} = (\text{N} - \text{N1B})\ \text{DETA2} \end{aligned}\right\} \ \text{N1B} \leq \text{N} \leq \text{N2B} \ , \\ &\text{IGAUSS} = 1. \end{aligned}$$

If IB $\leq$ 0, the responding surface is an initially flat plate. The subroutine is finished.

If IB > 0, data describing the cross section of the beam is needed. First, the subroutine reads ZU with FORMAT (3E10.2) and prints SLABL and ZU. It then reads WIDTHK, DZETAK, and ZETA(K) with FORMAT (3E10.2), sets W(K) = WIDTHK * DZETAK, and writes K, WIDTHK, DZETAK, ZETA(K), and W(K) for K = 1,2,...,LAYER.

The following table is an example for a 6-layer subdivision of the cross section of a solid, right-circular, cylindrical rod of unit radius:

K	WIDTHK	DZETAK	ZETA(K)
1	1.032494026	1/3	-0.8253794582
2	1.717575183	1/3	-0.4952276749
3	1.962319769	1/3	-0.1650758916
4	1.962319769	1/3	0.1650758916
5	1.717575183	1/3	0.4952276749
6	1.032494026	1/3	0.8253794582

To produce this table we first arbitrarily chose layers of equal thickness, DZETAK = 1/3. Then, we found WIDTHK so that W(K) = DZETAK * WIDTHK would equal the area of the circular cross section in that layer. Then we set $Z_K = K/3 - 7/6$ as a tentative location in the center of each layer. Finally we found ZETA(K) = c Z_K where $c^2 \sum (Z_K)^2 W(K) = \pi/4$, the area moment of inertia for a circle of unit radius.

Similarly, for a rectangular beam with unit thickness and width w:

K	WIDTHK	DZETAK	ZETA(K)	Z_K
1	w	1/6	-0.4225771	-5/12
2	w	1/6	-0.2535462	-3/12
3	w	1/6	-0.0845154	-1/12
4	w	1/6	0.0845154	1/12
5	w	1/6	0.2535462	3/12
6	w	1/6	0.4225771	5/12

(Choosing integration stations and weights for Gaussian integration, see Section 3.4, would probably be better for a rectangular beam.)

The choice of layers of equal thickness is convenient and reasonable for these two examples, but not for all cases. With an I-beam, for example, it would be more reasonable, and convenient, to approximate each flange with a layer and put four layers in the web. If the beam is not symmetric vertically it may be difficult to assign values that produce the correct area moment of inertia without altering the area, or the location of the centroid.

4.1.2 INGEOM for a Cylindrical Shell. The subroutine INGEOM for a cylinder is very simple (see the tabulation in Appendix F). The input is CYLL, RADIUS with FORMAT (2E12.6), where CYLL is the cylinder length and RADIUS is the radius to the midsurface. The subroutine sets

```
DETA1 = 2 π RADIUS
DETA2 = CYLL/FLOAT(NMESH)
IB = 0
R(N) = RADIUS
Z(N) = (N - 2) * DETA2
```

If Gaussian integration is desired, IGAUSS = 1.

4.2 PRESS

Subroutine PRESS supplies the pressure, P(N), at each mesh point that moves ($N1V \leq N \leq N2V$). (this is actually the pressure difference for shells and the normal force per unit length for beams, but we will continue to call it pressure.) The sign of P(N) was chosen so that a positive P(N) tends to crush a cylinder or push a flat plate down.

Most of the PRESS subroutines for RPSL1D have been in two sections. The first section reads input and makes other preliminary calculations the first time the subroutine is called. The other part supplies the P(N) each time the subroutine is entered.

PRESS is called in the initial portion of RPSL1D if LOAD = 0. It is called at the start of each succeeding time cycle in RPSL1D if LPRESS $\geq$ NCYCLE. Also, as long as LPRESS $\geq$ NCYCLE, the P(N) are multiplied by $a^{\frac{1}{2}}$ in subroutine DGEOM. This multiplication changes the P(N) from pressure to force per unit initial area. If the program is continued beyond cycle NCYCLE = LPRESS, the values of P(N) do not change. This means the force per unit initial area is fixed, the pressure varies with the change in area. (At the initiation of

damping, when NCYCLE = MDAMP, subroutine DAMP sets LPRESS and all P(N) to zero. This must be changed if the program is to continue with both pressure and damping.) NCYCLE is the count of time cycles. LOAD, LPRESS, and MDAMP are input parameters. If LOAD = 0 and LPRESS = 0, PRESS will never be called and all P(N) will be zero.

One of the weak points of using deformation codes to simulate reality is in accurately describing dynamic pressure loading. It is difficult to obtain accurate, repeatable measurements for enough points even with well-controlled experiments. Such measurements tend to be very oscillatory due to both reality and instrumentation. Even assuming a completely accurate pressure record at some point, the effective pressure on a thin shell would be different because of its rapid reaction. The best we can hope for is an approximate simulation that will lead to useful results. Three examples are discussed.

4.2.1 PRESS for Constant Pressure. This is the PRESS subroutine with the program in Appendix C. In the initial entrance for each run, the constand pressure, PO, is read with FORMAT (E12.6). During each entrance it sets P(N) = PO for N1V $\leq$ N $\leq$ N2V.

4.2.2 PRESS for Pressure as a Function of Time. This subroutine, called LINPRS, is tabulated with Example 1 in Appendix E. Pressure is a tabular function of time. On the first entrance, this subroutine reads pairs of time and pressure data into TPR(I) and PPR(I) with FORMAT (2E10.3) until TPR(I) < 0. Then IPRESS, the number of data points in the table, is set to I-1. (It is assumed that IPRESS $\leq$ 50, TPR(1) = 0.0, TPR(I+1) > TPR(I), and TPR(IPRESS) is larger than any time to be reached.

Linear interpolation is used at every entry to compute the pressure, PO, corresponding to time, TIME. The program then sets P(N) = PO for N1V $\leq$ N $\leq$ N2V.

4.2.3 PRESS for Two Phase Pressure Decay in a Cylinder. This subroutine, labeled PRESS of 3/10/76, is listed in Appendix F. It simulates the pressure in a cylinder from an explosion at its center. This rather involved subroutine is included because it demonstrates a number of the common features of pressure subroutines that simulate the loading from explosions. The unbalanced pressure P_N at point N is zero until the shock front arrives at time TA_N with peak pressure PO_N. The pressure then decays rapidly. In this subroutine it decays rapidly, exponentially until it reaches a pressure $\bar{P}$ which we call the quasi-static pressure. It then continues to decay exponentially at a much slower rate.

$$\begin{aligned} P_N &= 0.0 && \text{if } t < TA_N , \\ P_N &= PO_N\, e^{-\alpha(t - TA_N)} && \text{if } TA_N \leq t \leq TB_N , \\ P_N &= \bar{P}\, e^{-\beta(t - TB_N)} && \text{if } TB_N \leq t , \end{aligned} \tag{70}$$

where α and β are the decay factors and $\bar{P}$ is the quasi-static pressure which is reached at point N at time TB_N found from

$$\bar{P} = PO_N \, e^{-\alpha(TB_N - TA_N)} \, .$$

An outline of the subroutine is given below. First, the following input cards are read on the initial entry.

Content	FORMAT
$\bar{P}$, β, (PID(I), I=1,6)	(2E10.3,6A10)
CZ, CT, CP, TD	(4E10.3)
α, ---, ISM	(E10.3,10X,I10)
(ZSI(I), TSI(I), PSI(I), I=1,ISM)	(3E10.3)

Here,

$\bar{P}$ is the quasi-static pressure.

β is the quasi-static decay factor.

PID(I) is an alphanumeric title.

CZ, CT, CP, and TD are conversion factors to transform data to the desired units (see below).

α is the shock pressure decay factor.

ISM is the number of entries in the following table (ISM $\leq$ 50).

ZSI(I) is the distance from the middle, end 1, of the cylinder.

TSI(I) is the shock arrival time at ZSI(I).

PSI(I) is the peak shock pressure at ZSI(I).

These data cards are read, and their images printed, during the initial entrance. The tabular data is then converted to the desired units with the replacement formulas

$$ZSI(I) = CZ * ZSI(I) \, , \tag{71}$$

$$TSI(I) = CT * (TSI(I) - TD) \, , \tag{72}$$

$$PSI(I) = CP * PSI(I) \, . \tag{73}$$

This converted table is also printed. For N1V $\leq$ N $\leq$ N2V, TS(N) $\equiv TA_N$ and PS(N) $\equiv PO_N$ are found by linear interpolation at distance Z(N) = (N - 2) * DETA2. (Here is an example where the material parameter ξ is assumed to be length from end 1. Symmetry is assumed at end 1 where N = N1B = N1V = 2.) TSB(N) = TB_N is found from

$$TSB(N) = TS(N) + \{\ln(PS(N)/\bar{P})\}/\alpha \, . \tag{74}$$

The program then forms PSB(N) and transforms PS(N) to a more convenient form, as follows:

$$PS(N) = PS(N)\ e^{\alpha TS(N)} \tag{75}$$

$$PSB(N) = \bar{P}\ e^{\beta TSB(N)}\ . \tag{76}$$

This completes the preparation. For every entry, the subroutine computes $e^{-\alpha t}$ and $e^{-\beta t}$ and then, for $N1V \leq N \leq N2V$,

$$P(N) = 0.0 \qquad \text{if } t < TS(N)\ ,$$

$$P(N) = -e^{-\alpha t}\ PS(N) \equiv -PO_N\ e^{-\alpha(t - TA_N)} \qquad \text{if } TS(N) \leq t \leq TSB(N)\ , \tag{77}$$

$$P(N) = -e^{-\beta t}\ PSB(N) \equiv -\bar{P}\ e^{-\beta(t - TB_N)} \qquad \text{if } TSB(N) < t\ .$$

4.3 INVEL

Subroutine INVEL supplies the initial velocity of moving mesh points. The velocity of all mesh points is preset to zero before INVEL is called. This subroutine sets

$$DR(N) = \dot{R}\ , \tag{78}$$

$$DZ(N) = \dot{Z}\ , \tag{79}$$

where the velocity at point N is

$$\underset{\sim}{v} = \dot{R}\ \underset{\sim}{i}_3 + \dot{Z}\ \underset{\sim}{i}_2\ . \tag{80}$$

INVEL is called during the initiation sequence from the main program, RPSL1D, if parameter LOAD is less than or equal to zero. The components of initial velocity are changed to displacement increments by multiplication with Δt in RPSL1D.

4.3.1 INVEL for Normal Velocity at Specified Points. The INVEL subroutine catalogued with RPSL1D (see Appendix C) is a one-dimensional version of the INVEL tabulated in the User's Manual (Reference 2). This subroutine assumes all initial velocities are normal to the surface. The input cards are:

Cards	FORMAT
--,--,NI,NF,VR,NV	(10X,2I5,E12.6,I5)
--, N, V	(5X,I5,E12.6)

NI and NF are the minimum and maximum mesh numbers for an array of points to be given initial velocity VR. NV is the number of mesh points to receive initial velocity different from zero or VR. N is the mesh number for a point receiving velocity V.

Note! This subroutine was written to use input cards for the standard REPSIL. The mesh numbers NI, NF, and N are all relative to the mesh point at end 1 being number 1. In RPSL1D, the mesh point at

end 1 is numbered N1B. (Subroutine START sets N1B = 2.) Therefore, this INVEL subroutine adds N1B-1 (i.e. one) to all input mesh numbers.

The first input card is read and the program sets

$$DR(N) = -SNR(N) * VR \ , \tag{81}$$

$$DZ(N) = -SNK(N) * VR \ , \tag{82}$$

for NI + N1B-1 $\leq$ N $\leq$ NF + N1B-1, where the unit normal at mesh point N is

$$\underset{\sim}{n} = SNR(N)\ \underset{\sim}{i}_3 + SNK(N)\ \underset{\sim}{i}_2 \ .$$

(Notice the reversal in sign. VR is assumed to be an inwardly directed velocity, possibly from impulsive loading, on the outside surface.)

The program then reads NV cards of the second type (NV may be zero), replaces N by N + N1B-1, and sets

$$DR(N) = -SNR(N) * V \ , \tag{83}$$

$$DZ(N) = -SNK(N) * V \ . \tag{84}$$

The subroutine prints the initial velocities assigned. These initial velocities are later converted to displacement increments by multiplying them by Δt.

4.3.2 INVEL for Constant Lateral Velocity. This subroutine has been used to simulate a rod striking a wall. The changes in the catalogued INVEL to produce this are tabulated with Example 2 in Appendix E.

The constant velocity, VR, is read with effective FORMAT (20X,E12.6) and the subroutine sets

$$DZ(N) = -VR \tag{85}$$

for N1V $\leq$ N $\leq$ N2V.

4.3.3 INVEL for Beam Vibration in the Fundamental Transverse Mode. A number of INVEL subroutines have been coded for beams. They were either very simple or very specialized. As examples of simple codes for initially flat beams, consider the following for the fundamental mode of free transverse vibrations. The transverse velocity of any point x at time t may be written

$$\dot{R}(x,t) = A\ X(x)\ \cos(\omega t) \tag{86}$$

for these examples. The velocity at time t = 0 is

$$\dot{R}(x,0) = A\ X(x) \ . \tag{87}$$

Here ω is the fundamental frequency, X(x) is a function of the end conditions, and the parameter A is arbitrary. We assume evenly spaced points to find x. (These formulas are similar to those given by Enrico Volerra, E. C. Zachmanoglou, Dynamics of Vibration, Charles E. Merrill Books, Inc., Columbus, Ohio.)

Cantilever Beam:

$$x = 1.875104069\ (N - N1B)/(N2B - N1B)\ , \tag{88}$$

$$DR(N) = A\{\cosh(x) - \cos(x) - 0.7340955137\ [\sinh(x) - \sin(x)]\}\ . \tag{89}$$

Simply Supported Beam:

$$x = \pi(N - N1B)/(N2B - N1B)\ , \tag{90}$$

$$DR(N) = A\ \sin(x)\ . \tag{91}$$

Clamped Beam:

$$x = 4.730040744862\ (N - N1B)/(N2B - N1B)\ , \tag{92}$$

$$DR(N) = A\ \{\cosh(x) - \cos(x) - 0.9825022145\ [\sinh(x) - \sin(x)]\}\ . \tag{93}$$

4.4 INNORM

Subroutine INNORM is a new user-supplied subroutine that assigns the unit normal at clamped ends. It is called in START immediately after the call of INGEOM. The version of INNORM catalogued with the program in Appendix C is the only one used up to now. It produces the correct normal if the shape of the reference curve at the end can be written as a quadratic equation in ξ through the last three end points.

The components of the unit normal, $\underset{\sim}{n}$, are defined in terms of partial derivatives of the position vector $\underset{\sim}{r}$.

$$\underset{\sim}{r} = R\ \underset{\sim}{i}_3 + Z\ \underset{\sim}{i}_2\ , \tag{94}$$

$$R2 = \partial R/\partial\xi\ , \tag{95}$$

$$Z2 = \partial Z/\partial\xi\ , \tag{96}$$

$$D = \{(R2)^2 + (Z2)^2\}^{\frac{1}{2}}\ , \tag{97}$$

$$SNR(N) = Z2/D\ , \tag{98}$$

$$SNK(N) = -R2/D\ , \tag{99}$$

$$\underset{\sim}{n} = SNR(N)\ \underset{\sim}{i}_3 + SNK(N)\ \underset{\sim}{i}_2\ . \tag{100}$$

The partial derivatives, R2 and Z2, at the clamped ends are approximated by non-central differences through three end points. N is N1B or N2B, depending on which end is clamped.

$$\left(\text{e.g. } \left.\frac{\partial R}{\partial\xi}\right|_{N=N1B} = \{3R(N2B) - 4R(N2B - 1) + R(N2B - 2)\}/(2\Delta\xi)\ .\right) \tag{101}$$

4.5 ENDFRC

Subroutine ENDFRC is a user subroutine that supplies the forces and moments imposed at the ends of a beam. It is really a part of the option called APLFRC (Section 5.4) which was programmed to utilize this type of loading. The subroutine is not needed unless the APLFRC option is included in the program and either IBCE1 = 4 or IBCE2 = 4. With the APLFRC option, IBCE1 = 4 signals an applied force and moment at end 1 and IBCE2 = 4 signals an applied force and moment at end 2. A free end is a special case for which the applied force and moment are zero.

Subroutine ENDFRC is called from subroutine RESULT when N = N1B if IBCE1 = 4, or when N = N2B if INCE2 = 4 and IBCE1 $\neq$ 4. Since RESULT is not called in the initial sequence, there are no initial forces on the ends.

When ENDFRC is called, the force components and moments are supplied. The force at end 1 is

$$\text{EFR1}\ \underset{\sim}{i}_3 + \text{EFZ1}\ \underset{\sim}{i}_2 \ . \tag{102}$$

The moment is EM1. Similarly at end 2, the force components and moment are EFR2, EFZ2, and EM2, respectively. Figure 4.5 and the following list both explain the sign conventions.

EFR1 > 0 decreases R(N1B) ,

EFZ1 > 0 decreases Z(N1B) ,

EM1 > 0 decreases R(N1B) and increases R(N1B + 1) ,

EFR2 > 0 increases R(N2B) ,

EFZ2 > 0 increases Z(N2B) ,

EM2 > 0 increases R(N2B - 1) and decreases R(N2B) .

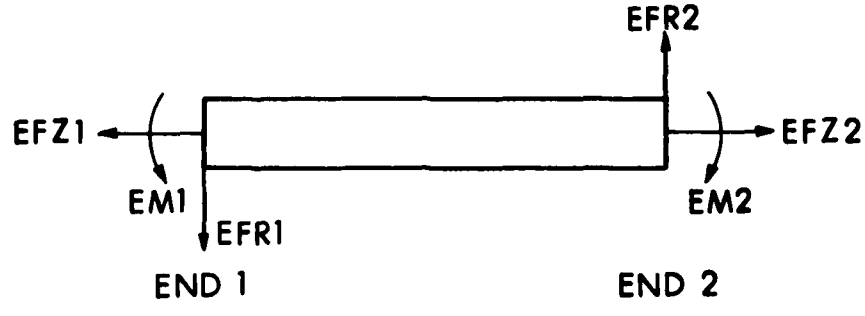


Figure 4.5 Sign Convention for Applied Forces on the End of a Beam.

The program outside ENDFRC does not change these terms. Constant values may be inserted in the initial entry and need not be reset. It does no harm to insert values at an end that is not forced (IBCE1 ≠ 4 or IBCE2 ≠ 4); the program ignores them.

4.5.1 ENDFRC for One or Both Ends Free. The initial entry into this ENDFRC sets EFR1, EFZ1, EM1, EFR2, EFZ2, and EM2 to zero. These are never changed. The APLFRC option then imposes a free end at end 1 if IBCE1 = 4, and/or a free end at end 2 if IBCE = 4. This subroutine is tabulated in Appendix F.

4.5.2 ENDFRC for Tabular Forces. This ENDFRC subroutine assumes that end 2 is free and the forces on end 1 are supplied by tables. On the initial entry this subroutine reads three tables. Each has a heading giving the number of entries and an alphanumeric title.

Cards	FORMAT
NFZPTS, (LABELV(I), I=1,7)	(I10,7A10)
(EFZT(I), TMFZ(I), I=1,NFZPTS)	(2E15.7)
NFTHS, (LABELV(I), I=1,7)	(I10,7A10)
(EFTHT(I), TMTH(I), I=1,NFTHS)	(2E15.7)
NFMUS, (LABELV(I), I=1,7)	(I10,7A10)
(EFMUT(I), TMMU(I), I=1,NFMUS)	(2E15.7)

The first table is the horizontal force component, EFZ1, as a function of time. The other two tables are of parameters θ and μ, respectively, as functions of time.

The initial entrance into the subroutine reads and prints the input and sets EFZ2 = EFR2 = AM2 = 0. On each entry, the subroutine finds EFZ1, θ, and μ as functions of time by linear interpolation. It then sets

$$EFR1 = \mu EFZ1 \ , \tag{103}$$

$$EM1 = -ZU \{\sin(\theta)\ EFZ1 + [1 - \cos(\theta)]\ EFR1\} \ . \tag{104}$$

4.5.3 ENDFRC for Tabular Forces (Modified). This subroutine is like the previous one except that the force on end 1 is assumed relative to the tangent, $\underset{\sim}{\tau}$, at end 1,

$$\underset{\sim}{\tau} = SNK(N1B)\ \underset{\sim}{i}_3 - SNR(N1B)\ \underset{\sim}{i}_2 \ , \tag{105}$$

rather than the horizontal direction, $\underset{\sim}{i}_2$. The force components for that subroutine are replaced by

$$EFR1 = EFR1 * SNR(N1B) - EFZ1 * SNK(N1B) \ , \tag{106}$$

$$EFZ1 = EFR1 * SNK(N1B) + EFZ1 * SNR(N1B) \ . \tag{107}$$

A slightly modified version of this is tabulated with example 2 in Appendix E.

5. OPTIONAL CHANGES

As has been stated before, it seems that any particular project for RPSL1D requires some changes in the program. Ideally, these changes are restricted to the user subroutines, discussed in the previous section, which describe the geometry and loading. In this section we will briefly discuss the seven sets of optional coding that are used with the two examples in Appendix E. They are of varying utility. The plotting option PLOTP is usually wanted. It was not added to the catalogued program because it is very short and usually requires some additional coding to select the functions of time to be plotted. The APLFRC option which permits loading the ends of beams by force and moment has been used a number of times. If it is used, the option SHR3/1 must be included and the energy computation EAPFRC should be added.

It is hoped that these options will be useful and serve as a guide for introducing others in the future.

5.1 PLOTP (Plot Functions of Time)

This coding was originally inserted to plot the force/unit initial area (force/unit initial length, for beams) at prescribed mesh points as functions of time. However, any function of time may be plotted by inserting coding that stores that function's value into P(N) (N2B < N < 104) each cycle, and requesting a plot for that P(N) on the new input card, card 11a. If the PLOTP option is in the deck, this input card must be included (see Section 6). It selects NNPE ($0 \leq NNPE \leq 9$) values of N for which plots of P(N) are to be made.

The changes to RPSL1D are rather trivial: some new terms are inserted into COMMON and MAIN, the new input card is read and its image is printed in START, and the requested P(N) are stored in the plot data file in PDATA. More extensive changes were required in the plotting program. These changes are included in the plotting program stored in file RPSL1DPLOT except for two orders (the optional coding) which read from the plot data file and activate P(N) plotting. This option is used with both examples in Appendix E. (In example 1 the total large core memory requested for plotting is greater than 131,071 words. Hence, the LCM = I parameter is specified on the FTN control card.)

5.2 SHR3/1 (Compute Moment, Shear and Axial Forces

This option is for beams only. It computes the moment, M, at each mesh point and computes the shear force, V, and the axial force, Q, between mesh points. The following equations were extracted from Reference 1 by Dr. Santiago.

$$M = a\,\hat{M}^{22} \tag{108}$$

$$Q = a\,(\hat{Q}^{22} - b_2^2\,\hat{M}^{22}) \tag{109}$$

$$V = (1/a^{\frac{1}{2}})\;\partial(a\,\hat{M}^{22}/\partial\xi^2) \tag{110}$$

The RPSL1D analog of these equations is

$$AMS(N) = A22 * F22 \tag{111}$$

$$QS(N) = A22 * (Q22 - BM22 * F22) \tag{112}$$

$$VS(N) = \{AMS(N+1) - AMS(N)\}/(DETA2 * SRA)\ . \tag{113}$$

For convenience, the three new arrays are put in COMMON in COMDECK MAIN. The rest of this option is in subroutine RESULT. A table of N, AMS(N), VS(N), QS(N), A22, B22, Q22, and F22 may be printed at cycles prescribed by coding inserted with the option.

5.3 MSQSVS (Plot Moment, Shear and Axial Forces)

This option was inserted to plot the moment, shear force, and axial force computed with option SHR3/1 on the same cycles that cross-sectional plots are made (i.e. whenever NCYCLE = NC3DP(I)). The only change in RPSL1D is one statement in PDATA that stores the three arrays AMS(N), QS(N), and VS(N) on the plot data file. The bulk of this option is inserted into the plotting program catalogued in file RPSL1DPLOT.

5.4 APLFRC (Applied Forces at Beam Ends)

This option was inserted to permit an applied vector force and moment at either or both ends of a beam. We will refer to this as a forced end. A free end is a special case with zero applied force and moment. It is assumed that the option SHR3/1 is included in the program.

The equation of motion, given in Reference 1, reduced to that of a beam with no external forces is

$$\Delta\underset{\sim}{r}^{+} = \Delta\underset{\sim}{r} + \left\{\hat{\underset{\sim}{M}}^{*22}{}_{,22} + \hat{\underset{\sim}{N}}^{*2}{}_{,2}\right\}(\Delta t)^2/(A^{\frac{1}{2}}\Gamma_0)\ , \tag{114}$$

where

$$\hat{\underset{\sim}{M}}^{*22} = a^{\frac{1}{2}}\hat{M}^{22}\,\underset{\sim}{n} \tag{115}$$

and

$$\hat{\underset{\sim}{N}}^{*2} = a^{\frac{1}{2}}\,(\hat{Q}^{22}\,\underset{\sim}{a}_2 + \Gamma_{22}{}^2\,\hat{M}^{22}\,\underset{\sim}{n}) \tag{116}$$

are vectors defining the bending and stress resultant terms. The notation $X_{,2}$ and $X_{,22}$ denotes the partial derivatives $\partial X/\partial\xi$ and $\partial^2 X/(\partial\xi)^2$. The mass per initial unit length for a beam is given by $A^{\frac{1}{2}}\Gamma_0 = A^{\frac{1}{2}}\rho\sum W_k$.

From the equations for shear force, axial force, and moment, respectively,

$$V = a^{\frac{1}{2}} (\hat{M}^{22}{}_{,2} + 2\Gamma_{22}{}^{2} \hat{M}^{22} = (a \hat{M}^{22})_{,2}/a^{\frac{1}{2}} , \tag{117}$$

$$Q = a (\hat{Q}^{22} - b_2^2 \hat{M}^{22}) , \tag{118}$$

$$M = a \hat{M}^{22} , \tag{119}$$

and the relations

$$a^{\frac{1}{2}}{}_{,2} = \Gamma_{22}{}^{2} a^{\frac{1}{2}} , \tag{120}$$

$$\underset{\sim}{n}_{,2} = -b_2^2 \underset{\sim}{a}_2 , \tag{121}$$

$$\underset{\sim}{a}_2 = a^{\frac{1}{2}} \underset{\sim}{\tau} , \tag{122}$$

which are also taken from Reference 1, we can transform the vector

$$(\hat{\underset{\sim}{M}}^{*22}{}_{,22} + \hat{\underset{\sim}{N}}^{*2}{}_{,2}) \tag{123}$$

into the form

$$(V \underset{\sim}{n} + Q\underset{\sim}{\tau})_{,2} \tag{124}$$

(The shear force and axial force, V and Q, are computed at all midmesh points, and the moment, M, is computed at all mesh points, except at forced ends, by option SHR3/1.) Vector $\underset{\sim}{n}$ is the unit normal to the reference surface and $\underset{\sim}{\tau}$ is the unit tangent.

$$\underset{\sim}{n} = n_k \underset{\sim}{i}_2 + n_r \underset{\sim}{i}_3 . \tag{125}$$

$$\underset{\sim}{\tau} = n_r \underset{\sim}{i}_2 - n_k \underset{\sim}{i}_3 . \tag{126}$$

The force vector

$$\underset{\sim}{F} = (V \underset{\sim}{n} + Q\underset{\sim}{\tau}) \tag{127}$$

can be computed at all midmesh points if the moment $M = a \hat{M}^{22}$ is known at the mesh points.

The APLFRC option assumes a new user subroutine, ENDFRC (see Section 4.5), which supplies M and $\underset{\sim}{F}$ at forced ends. In the following, we use the subscript N to denote quantities at the N'th mesh point, N = N1B at end 1 and N2B at end 2, and subscript N+½ to denote midmesh quantities. If end 1 is forced,

$$\underset{\sim}{F}_{N1B} = EFZ1 \underset{\sim}{i}_2 + EFR1 \underset{\sim}{i}_3 , \; M_{N1B} = EM1 . \tag{128}$$

If end 2 is forced,

$$\underset{\sim}{F}_{N2B} = EFZ2\ \underset{\sim}{i}_2 + EFR2\ \underset{\sim}{i}_3\ ,\quad M_{N2B} = EM2\ . \tag{129}$$

For N1B $\leq$ N < N2B,

$$\underset{\sim}{F}_{N+\frac{1}{2}} = (V_{N+\frac{1}{2}}\ \underset{\sim}{n}_{N+\frac{1}{2}} + Q_{N+\frac{1}{2}}\ \underset{\sim}{\tau}_{N+\frac{1}{2}}) \tag{130}$$

Then, the force per unit length at N = N1B and N1B+1 if end 1 is forced, and at N = N2B and N2B-1 if end 2 is forced, is found by the difference formulas:

$$(\underset{\sim}{F},_2)_{N1B} = (\underset{\sim}{F}_{N1B+\frac{1}{2}} - \underset{\sim}{F}_{N1B})/(\Delta\xi/2)\ , \tag{131}$$

$$(\underset{\sim}{F},_2)_{N1B+1} = (\underset{\sim}{F}_{N1B+3/2} - \underset{\sim}{F}_{N1B+\frac{1}{2}})/(\Delta\xi)\ , \tag{132}$$

$$(\underset{\sim}{F},_2)_{N2B-1} = (\underset{\sim}{F}_{N2B-\frac{1}{2}} - \underset{\sim}{F}_{N2B-3/2})/(\Delta\xi)\ , \tag{133}$$

$$(\underset{\sim}{F},_2)_{N2B} = (\underset{\sim}{F}_{N2B} - \underset{\sim}{F}_{N2B-\frac{1}{2}})/(\Delta\xi/2)\ . \tag{134}$$

The first and last of these are not central differences. The form

$$(\underset{\sim}{F},_2)_N = (\underset{\sim}{M}^{*22},_{22} + \underset{\sim}{N}^{*2},_2)\ , \tag{135}$$

where only central differences are employed, is used at all other moving mesh points (see Section 3.2).

This option includes minor changes in MAIN, START, BOUNDR, and BOUNDU, and more extensive changes in GRAD, RESULT, and MOTION, plus the user supplied subroutine ENDFRC. Option SHR3/1 must also be used, and option EAPFRC should be.

5.5 EAPFRC (Energy for APLFRC)

In the section for APLFRC we described the option which allows forces to be applied at either end of a beam. The optional coding labeled EAPFRC, which computes the work on the beam by the applied forces and moments, is properly a part of the APLFRC option.

The work by a directed force, $\underset{\sim}{F}$, applied at a point that moves a vector distance $\Delta\underset{\sim}{r}$ is

$$\Delta W_F = \underset{\sim}{F} \cdot \Delta\underset{\sim}{r}\ . \tag{136}$$

The work by a moment of magnitude M applied while the normal vector changes by an angle $\Delta\theta$ is

$$\Delta W_M = \pm M\ \Delta\theta\ . \tag{137}$$

(The sign in this equation depends on the sign convention used.)

This work must be compatible with work from pressure loading (see Reference 2, pages 45 and 46), so an analogous method is used in RPSL1D. That is, the external work from time t-Δt to time t, $\Delta W_F(t-\frac{1}{2}\Delta t)$ and $\Delta W_M(t-\frac{1}{2}\Delta t)$, will be the average of "work increments" computed at times t-Δt and t. We compute:

$$\Delta W_F(t) = \underset{\sim}{F}(t) \cdot \{\Delta \underset{\sim}{r}(t-\tfrac{1}{2}\Delta t) + \Delta \underset{\sim}{r}(t+\tfrac{1}{2}\Delta t)\}/2 , \tag{138}$$

$$\Delta W_F(t-\tfrac{1}{2}\Delta t) = \{\Delta W_F(t-\Delta t) + \Delta W_F(t)\}/2 , \tag{139}$$

$$\Delta W_M(t) = - M(t) \quad \{\Delta\theta(t-\tfrac{1}{2}\Delta t) + \Delta\theta(t+\tfrac{1}{2}\Delta t)\}/2 , \tag{140}$$

$$\Delta W_M(t-\tfrac{1}{2}\Delta t) = \{\Delta W_M(t-\Delta t) + \Delta W_M(t)\}/2 , \tag{141}$$

at the forced ends.

Except for some preliminary set up, all the computation is carried out in MOTION. The values of $\Delta W_F(t-\Delta t)$, $\Delta W_M(t-\Delta t)$, $\Delta\theta(t-\frac{1}{2}\Delta t)$, and the components of $\Delta \underset{\sim}{r}(t-\frac{1}{2}\Delta t)$ at any forced end, is saved from the previous cycle. The components of $\Delta \underset{\sim}{r}(t+\frac{1}{2}\Delta t)$ are already stored in the arrays DR(N) and DZ(N) by MOTION. We need to compute $\Delta\theta(t+\frac{1}{2}\Delta t)$. We define $\Delta\theta(t+\frac{1}{2}\Delta t)$ by

$$\sin\{\Delta\theta(t+\tfrac{1}{2}\Delta t)\}\ \underset{\sim}{i}_1 = \underset{\sim}{n}(t) \times \underset{\sim}{n}(t+\Delta t) . \tag{142}$$

The unit normal at time t is available,

$$\underset{\sim}{n}(t) = SNK(N)\ \underset{\sim}{i}_2 + SNR(N)\ \underset{\sim}{i}_3 , \tag{143}$$

but $\underset{\sim}{n}(t+\Delta t)$ must be computed. This is computed from

$$\underset{\sim}{a}_1 = \underset{\sim}{i}_1 , \tag{144}$$

$$\underset{\sim}{a}_2 = \partial \underset{\sim}{r}^+/\partial\xi , \tag{145}$$

$$\underset{\sim}{n}(t+\Delta t) = (\underset{\sim}{a}_1 \times \underset{\sim}{a}_2)/|\underset{\sim}{a}_1 \times \underset{\sim}{a}_2| , \tag{146}$$

where

$$\underset{\sim}{r}^+ = \underset{\sim}{r}(t) + \Delta \underset{\sim}{r}(t+\tfrac{1}{2}\Delta t) \tag{147}$$

and $\underset{\sim}{a}_2$ is approximated by a forward or backward difference.

5.6 EROD (Erosion at a Beam End)

This option depends on the applied force option, APLFRC, being in RPSL1D. The basis for our model is to assume an end mesh point, denoted by N = N1B, which moves according to forces on the true, nearby, eroding end. Whenever enough erosion has occurred, the end mesh point is shifted

to the next mesh point. (The eroding end cannot approach the midpoint of the first mesh too closely, or the difference equations at that end would be unreliable.)

We inserted a new subroutine, ERODE, which supplies the distance, DE, from the mesh point N1B to the eroding end and advances the endpoint indices when necessary. DE is zero initially. Whenever DE is greater than $\Delta\xi/4$, we subtract $\Delta\xi$ from DE and add 1 to N1B, N1V, and N1A (these are all indices for the end point). The difference equations for the shear in RESULT and force/unit length on the end point in MOTION were changed to reflect the position of the eroding end. The ENDFRC subroutine was altered to use quadratic interpolation to find the normal at the eroding end. (This last change was not really tested since the rod remained straight in our application.) Several minor inserts were made to permit reasonable plots.

This option is not as well tested as other features of RPSL1D. This brief discussion is included because the option is used with Example 2 (Appendix E). The motivation for Example 2 was to demonstrate that the RPSL1D program could reproduce experimental strain records at two points along a long rod penetrator, assuming a stress-strain curve, and a table of force on the end and position of the end as functions of time, all derived from the Karmann-Taylor theory[3]. From the two strain records, the density, and the cross-sectional area of the rod, one can derive the stress and force for any strain, and the velocity at which the strain is propagated. From this velocity, one has a linear relation between initial time and position of the generating force for each strain. Two quite different tables of erosion and force as functions of time were generated, one for a slow erosion rate and the other for a rapid erosion rate. Both were used as input for RPSL1D, and both produced strain records in fair agreement with the test data.

5.7 BSTRS (Stress-Strain Option for Beams)

The optional coding labelled BSTRS is a stress-strain routine for beams, which models the uniaxial curve more smoothly than the mechanical sublayer model by using more points. The mechanical sublayer model in the catalogued routine is generally superior, but the number of segments in the polygonal stress-strain curve must be restricted.

In the erosion problem discussed in Section 5.6, we wanted to compare the experimental strain-time response in a bar with output from RPSL1D. The strain response from RPSL1D at points along a bar agree closely with those predicted by the Karmann-Taylor theory. That is, the velocity at which a particular strain level is propagated down the bar is proportional to the square root of the slope of the stress-strain curve at that strain. Therefore, the strain-time records from RPSL1D at points away from the end of the rod, in response to an increasing force at the end, have near discontinuities corresponding to the corners of the polygonal stress-strain curve where there are discontinuities in the slope. To remove these discontinuities, one needs

---

[3] *A.D. Gupta and J.D. Wortman, "An Eroding Long Rod Penetrator Model for Hard Target Penetration," Proceedings of the Third ASCE/EMD Specialty Conference, September 17-19, 1979, University of Texas, Austin, Texas, pp 714-717.*

a smoother stress-strain curve. This was impractical with the mechanical sublayer model, so the BSTRS option was coded. If enough points are used, the uniaxial loading curve will be smooth enough. This option is not programmed for a strain rate effect nor for biaxial loading and there is no obvious simple way to include either of them.

The positive portion of the uniaxial stress-strain curve is entered as a table of points, $(\varepsilon_i, \sigma_i)$, with FORMAT (2E15.7), which terminates with $\varepsilon_{NEST} < 0.0$ (NEST $\leq$ 100). Then ε_{NEST} is replaced by 1000.0 (infinity) and σ_{NEST} is replaced by σ_{NEST-1}. As before, for the mechanical sublayer model, the point $(\varepsilon_1, \sigma_1)$ is replaced by (SIGZ/E, SIGZ), the elastic limit. Also, as before, these are not engineering stress nor strain. If $(\varepsilon_{ei}, \sigma_{ei})$ are engineering values,

$$\varepsilon_i = \varepsilon_{ei}(1 + \tfrac{1}{2}\varepsilon_{ei}) \ , \tag{148}$$

$$\sigma_i = \sigma_{ei}(1 + \sigma_{ei}) \ . \tag{149}$$

These correspond to the mixed components ε_2^2, σ_2^2 of the strain and stress tensors used in RPSL1D. The negative portion of the curve is defined by $\sigma(-\varepsilon) = -\sigma(\varepsilon)$.

For each integration station at each mesh point and at each midmesh, we keep track of the strain, $\varepsilon = \varepsilon_2^2$, and the mean value of possible elastic strain variation, ε_m, in addition to $\sigma = \sigma_2^2$.

Initially we set $\varepsilon = \varepsilon_m = \sigma = 0.0$.

Assume we have ε, σ, and ε_m from time t-Δt (call these ε^-, σ^-, and ε_m^-), and $\Delta\varepsilon$, the change in ε from time t-Δt to time t. We recognize five cases for $\varepsilon_m^- \geq 0$. (For $\varepsilon_m^- < 0$, we change the sign on ε^-, σ^-, ε_m^-, and $\Delta\varepsilon$, and proceed as for $\varepsilon_m^- \geq 0$. At the end, we change the signs on ε, σ, and ε_m.)

In all five cases we set

$$\varepsilon = \varepsilon^- + \Delta\varepsilon \ . \tag{150}$$

Case	Conditions	Results
(1)	$\Delta\varepsilon \geq 0$	$\sigma = \sigma^- + E\,\Delta\varepsilon$
	$\varepsilon \leq \varepsilon_m + \varepsilon_1$	$\varepsilon_m = \varepsilon_m^-$
(2)	$\Delta\varepsilon < 0$	$\sigma = \sigma^- + E\,\Delta\varepsilon$
	$\varepsilon > \varepsilon_m - \varepsilon_1$	$\varepsilon_m = \varepsilon_m^-$

Case	Conditions	Results
(3)	$\Delta\varepsilon > 0$	$\sigma = \sigma(\varepsilon)$
	$\varepsilon > \varepsilon_m + \varepsilon_1$	$\varepsilon_m = \varepsilon - \varepsilon_1$
(4)	$\Delta\varepsilon < 0$	$\sigma = \sigma(\varepsilon + 2\varepsilon_1) - 2\sigma_1$
	$\varepsilon < \varepsilon_m - \varepsilon_1$	$\varepsilon_m = \varepsilon + \varepsilon_1$
	$\varepsilon \geqslant -\varepsilon_1$	
(5)	$\Delta\varepsilon < 0$	$\sigma = -\sigma(-\varepsilon)$
	$\varepsilon < \varepsilon_m - \varepsilon_1$	$\varepsilon_m = \varepsilon + \varepsilon_1$
	$\varepsilon < -\varepsilon_1$	

The functional value $\sigma(\varepsilon)$ is found from the table $(\varepsilon_i, \sigma_i)$ by linear interpolation.

(We should mention that the computation of strain energy is as if unloading to $\sigma = 0$ can take place entirely elastically. That is, the strain energy is assumed proportional to $(\sigma)^2$ where $\sigma = E\,\varepsilon$. This assumption, also used in REPSIL and the catalogued RPSL1D, is only true if $|\sigma| < 2\sigma_1$.)

The UPDATE cards for this option include some changes and additions to COMMON in COMDECK MAIN, a few orders to initiate some new arrays to zero in the main program, RPSL1D, a change to read and store the new form for the stress-strain table (input cards 7) in subroutine START, and a complete replacement of BMSTRS, the stress subroutine for beams.

6. INPUT

RPSL1D originally used the same input as the standard REPSIL (see Reference 2, pp 54-71). This is still true if the user-coded subroutines are compatible. Table 6.1 lists the input for RPSL1D. The first part of this table is like table 3.1 of Reference 2. Variables that are no longer used have a line drawn through them. A brief description for each variable is given after the table. More complete descriptions for some variables may be found in Reference 2. These descriptions are still valid and will not be repeated here. Units are indicated in brackets: T for time, L for length, and F for force. Limits are those for the catalogued RPSL1D (Appendix C); most of these can be easily increased.

The input for user subroutines is listed in the order the subroutines will be called. Examples of input are given, but the descriptions of the input variables for the user subroutines are purposely sketchy. The user must check these subroutines and their input. INGEOM is always called first and usually requires input from cards. INNORM is called next; no input has been needed so far. PRESS is called next if LOAD ≠ 0. INVEL is called

next if LOAD ≤ 0. (PRESS will be called next if LPRESS > 0; so far no version of PRESS has been coded for RPSL1D that required input after the initial entrance. The initial entrance should occur with LOAD $\neq 0$, but it could occur here.) ENDFRC will be called next if option APLFRC is used. Subroutine ERODE, part of option EROD, would be called last.

CARD	VARIABLES	FORMAT
1	TITLE	8A10
2	~~MESH~~, NMESH, LAYER, YLDFAC	3I5,E12.6
3	MAXC, NCONT, ~~WRITE~~, DELTAT	3I5,E12.6
4	~~IBCE1~~, IBCE2, ~~IBCE3~~, IBCE4	16I5
5	LOAD, LPRESS, MDAMP, DAMPF, DFACT	3I5,2E12.6
6	E, FNU, SIGZ, RHO, THICKN, NSFL, ISR	5E12.6,2I5
7	(SSIG(J), SEPS(J), DSR(J), PSR(J), J=1,NSFL)	4E15.7
8	NPRINT, (JCHK(I), I=1,3)	16I5
9	NUMCY, (NCYCH(I), I=1,NUMCY)	16I5
10	NLPRIN, (JCYNLP(I), I=1,NLPRIN)	16I5
11	N3D, (NC3DP(I), I=1,N3D)	16I5
11a	NNPE, (NPE(I), I=1,NNPE)	16I5
12	~~ETAD1~~, ETAD2, NSTRN	2E10.4,I5
13	(~~ETAG1(I)~~, ETAG2(I), ANGLE(I), ANDLB(I), NETAG(I), I=1,NSTRN)	4E10.4,I5
14	Input for subroutine INGEOM	
15	Input for subroutine INNORM	
16	Input for subroutine PRESS	
17	Input for subroutine INVEL	
18	Input for subroutine ENDFRC	
19	Input for subroutine ERODE	

Table 6.1 List of Input Cards

- IBCE4 from card 4 is stored in position IBCE1 and used as end 1 condition control.
- Omit card 7 if NSFL = 0 (or if NSFL = 1 and ISR = 0), unless the BSTRS option is used. Card 7 is read with a different form and FORMAT with option BSTRS.
- Card 11a is included if, and only if, the PLOTP option is used.

The input variables in table 6.1 are described below. Additional descriptions are given in Chapter 3 of Reference 2.

Card 1 TITLE — Alphanumeric statement for output identification.

Card 2 ~~MESH~~ — Not used.

NMESH — Number of meshes. (NMESH < 102)

LAYER — Number of integration stations through the thickness. This may be the number of layers as in REPSIL.

	YLDFAC	Parameter controlling substep size for plastic yielding in shells. (2.0 is suggested)
Card 3	MAXC	Final time step. (Total number of time steps to final time.)
	NCONT	Initial time step for run. This must be zero. Restart capability has been deleted from the CDC version.
	~~WRITE~~	Not used.
	DELTAT	Suggested time step. This will be replaced if it is larger than the computed stable Δt or if entered as 0.0. [T]
Card 4	~~IBCE1~~	Numbers for end conditions. IBCE1 and IBCE3 are unused.
	IBCE2	IBCE2 is for end 2 (ξ a maximum). IBCE4 is for end 1
	~~IBCE3~~	(ξ a minimum). IBCE4 is stored in FORTRAN position IBCE1).
	IBCE4	1 - clamped end, 2 - symmetry end, 3 - hinged end, 4 - free end, end 2 only with catalogued RPSL1D, forced end, either end with APLFRC option.
Card 5	LOAD	Parameter which controls calls of INVEL and PRESS in the initiation sequence. (If LPRESS > 0, LOAD ≠ 0) 1 - call PRESS (supplies initial pressure) 0 - call INVEL (supplies initial velocity) -1 - call PRESS and then call INVEL
	LPRESS	Subroutine PRESS is called each time cycle to supply pressures in P(N), and then P(N) is converted to force/initial unit surface area, until NCYCLE > LPRESS. The force/initial unit area is constant after NCYCLE > LPRESS. (For beams, unit area is replaced by unit length.)
	MDAMP	If NCYCLE ≥ MDAMP, the damping procedures are carried out. If NCYCLE = MDAMP, the program sets LOAD = 0, LPRESS = 0, and all P(N) = 0.
	DAMPF	Viscous damping coefficient. [FT/L^3]
	DFACT	Parameter controlling termination of program during damping. (Suggest .001)
Card 6	E	Young's modulus. [F/L^2]
	FNU	Poisson's ratio, ν.
	SIGZ	Yield stress, σ_o. [F/L^2] (See card 7.)
	RHO	Initial mass density per unit volume, ρ. [FT^2/L^4, (i.e. mass/L^3)]
	THICKN	Thickness of shell. [L]
	NSFL	Number of changes of slope in the polygonal approximation of the uniaxial loading curve. (NSFL < 5) 0 - elastic behavior, no plasticity. 1 - elastic perfectly plastic response, >1 - elastoplastic, strain hardening response.

	ISR	Strain rate sensitivity control. 0 - no strain rate effect, 1 - plasticity is strain rate dependent.
Card 7	SSIG(J), SEPS(J)	Stress, σ [F/L^2], and strain, ε [L/L], at the corners of the polynomial, uniaxial loading curve. (SSIG(1) is replaced by SIGZ and SEPS(1) by SIGZ/E.) (If (e_s,σ_s) are 'engineering' strain and stress, $\sigma = \sigma_s(1 + e_s)$, $\varepsilon = e_s(1 + e_s/2)$.)
	DSR(J), PSR(J)	Constants for strain rate behavior. These must be included if ISR = 1.

With the BSTRS option, cards 7 become:

(EEB(I), SSB(I), I = 1,2,...) FORMAT(2E15.7)

	EEB(I), SSB(I)	Table of strain and stress (reverse order from above) for up to 100 points. Table is terminated by EEB(NEST) < 0. SSB(1) is replaced by SIGZ, and EEB(1) by SIGZ/E, EEB(NEST) by 1000, and SSB(NEST) by SSB(NEST-1).
Card 8	NPRINT	Number of cycles between surface strain prints at points defined on cards 13. Maximum deflection and extreme strains are also printed.
	JCHK(I)	Numbers controlling printing at cycles on cards 9. JCHK(1), or JCHK(2), > 0 requests prints of coordinates, displacement increments, and pressure at mesh points. JCHK(3) > 0 requests prints of surface normals, and a print of surface strains on both surfaces, at all mesh points.
Card 9	NUMCY	Number of cycles for JCHK(I) controlled prints (card 8) and energy balance summary. (<51)
	NCYCH(I)	Cycles for prints.
Card 10	NLPRIN	Number of JCNLP(I) to follow. (<51)
	JCNLP(I)	Cycles to print IMAT(N,K) and IMATM(N,K) arrays.
Card 11	N3D	Number of cycles at which displacement plots are to be made. (<51)
	NC3DP(I)	Cycles for plots.
Card 11a	(Include if PLOTP option is used.)	
	NNPE	Number of plots to be made. ($0 \leq NNPE \leq 9$)
	NPE(I)	Index, N, of P(N) to plot as a function of time. P(N) ($N1V \leq N \leq N2V$) is force per initial unit area (initial length for beams). Other functions of time may be plotted by storing them in P(N) (N2B < N < 104) and listing N on this card.

Card 12 ~~ETAD1~~ Not used.

ETAD2 Material coordinate of location at which displacement as a function of time is to be plotted.

NSTRN Number of locations at which surface strains listed on cards 13 are to be plotted and printed (see card 8). (≤ 6)

Card 13 ~~ETAG1(I)~~ Unused.

ETAG2(I) Material coordinate for surface strains.

ANGLE(I), ANGLB(I) Angles for additional strain calculations. (See Figure 3.5, Reference 2)

NETAG(I) Selects surface for strain.
0 - Surface toward positive normal,
1 - Surface away from positive normal.

[Sample Input for User's Subroutines. See USER SUBROUTINE section.]

Card 14 [Subroutine INGEOM]

[INGEOM for semi-infinite flat plate or beam. Catalogued INGEOM]

(SLABL, SLABW, IB (2E10.5,I5))

SLABL Length.

SLABW Multiplier for energy computation, $\Delta\xi^1$ = 2.0 * SLABW. Not used for beams.

IB IB > 0 denotes a beam.

If IB > 0, also read

(ZU (E10.2))
((WIDTH(K), DZETA(K), ZETA(K), K = 1,...,LAYER) (3E10.2))

ZU Location of upper surface.

WIDTH(K) Width of K'th layer.

DZETA(K) Thickness of K'th layer.

ZETA(K) Integration station for K'th layer.

[INGEOM for a right circular cylinder]

(CYLL, RADIUS (2E12.6))

Card 15 [Subroutine INNORM]

(No input for present INNORM)

Card 16 [Subroutine PRESS]

[Constant pressure. Catalogued PRESS]

(PO (E12.6))

PO Pressure. Force per unit length for beams.

[Time dependent pressure, LINPRS]

```
((TPR(I), PPR(I), I=1,2,...)          (2E10.3))
((-1.0)                               (2E10.3))
```

[PRESS with tabular data, exponential decay. 3/10/76]

```
(PBAR, BETA, (PID(I), I=1,6)          (2E10.3,6A10))
(CONCZ, CONCT, CONCP, TDIF               (4E10.3))
(ALPHAS, NOUSE, ISM                   (2E10.3,I10))
((ZSI(I), TSI(I), PSI(I), I=1,ISM)       (3E10.3))
```

Card 17 [Subroutine INVEL]

[Catalogued INVEL. Like INVEL in Reference 2.]

```
(--, NI, NF, VR, NV          (10X,2I5,E12.6,I5))
(--, N, V                         (5X,I5,E12.6))
```

NI, NF Mesh points NI+1 through NF+1 are given initial normal velocity -VR. Mesh point at end 1 is numbered N1B = 2.

VR Velocity. [L/T]

NV Number of cards to follow.

N Mesh point N+1 given initial normal velocity -V.

V Velocity. [L/T]

Card 18 [Subroutine ENDFRC. Needed with APLFRC option.]

[End 2 free. Tabular forces on end 1.]

```
(NFZPTS, (LABELV(I), I=1,7)           (I10,7A10))
((EFZT(I), TMFZ(I), I=1,NFZPTS)         (2E15.7))
(NFTHS, (LABELV(I), I=1,7)            (I10,7A10))
((EFTHT(I), TMTH(I), I=1,NFTHS)         (2E15.7))
(NFMUS, (LABELV(I), I=1,7)            (I10,7A10))
((EFMUT(I), TMMU(I), I=1,NFMUS)         (2E15.7))
```

Card 19 [Subroutine ERODE. Part of EROD option.]

```
(NTEPTS, (LABEL(I), I=1,7)            (I10,7A10))
((TEROD(I), XEND(I), I=1,NTEPTS)        (2E15.7))
```

APPENDIX A

LIST OF EQUATIONS

In this section, we list the principal equations used in the standard REPSIL and the corresponding equations used for axial symmetry, slab symmetry, and beams, respectively. The notation in column 1 is that of Reference 1, except we have assigned an orthonormal cartesian vector basis ($\underset{\sim}{i}_1$, $\underset{\sim}{i}_2$, $\underset{\sim}{i}_3$) and defined corresponding Cartesian components of position, Y^1, Y^2, and Y^3.

The notation 'same' in any column means that the entry is the same as that in the column to the left.

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\underset{\sim}{x} = \underset{\sim}{x}(\xi^1, \xi^2, \zeta, t)$ $= \underset{\sim}{r}(\xi^1, \xi^2, t) + \zeta \underset{\sim}{n}(\xi^1, \xi^2, t)$	$\underset{\sim}{x} = \underset{\sim}{x}(0, \xi, \zeta, t) = \underset{\sim}{r} + \zeta \underset{\sim}{n}$ Evaluate at $\theta = 0$.	$\underset{\sim}{x} = \underset{\sim}{x}(Y^1, \xi, \zeta, t) = \underset{\sim}{r} + \zeta \underset{\sim}{n}$	$\underset{\sim}{x} = \underset{\sim}{x}(0, \xi, \zeta, t) = \underset{\sim}{r} + \zeta \underset{\sim}{n}$
$\underset{\sim}{r} = Y^k \underset{\sim}{i}_k = Y^1 \underset{\sim}{i}_1 + Y^2 \underset{\sim}{i}_2 + Y^3 \underset{\sim}{i}_3$ $Y^k = Y^k(\xi^1, \xi^2, t)$	$\underset{\sim}{r} = R(\xi, t)\, \hat{\underset{\sim}{r}}(0) + Z(\xi, t)\, \hat{\underset{\sim}{k}}$ $\hat{\underset{\sim}{r}} = \cos\theta\, \underset{\sim}{i}_3 + \sin\theta\, \underset{\sim}{i}_2 \; (= -d\hat{\underset{\sim}{\theta}}/d\theta)$ $\hat{\underset{\sim}{\theta}} = -\sin\theta\, \underset{\sim}{i}_3 + \cos\theta\, \underset{\sim}{i}_2 \; (= d\hat{\underset{\sim}{r}}/d\theta)$ $R = ((Y^3(0, \xi, t))^2 + (Y^1(0, \xi, t))^2)^{1/2}$ $= Y^3(0, \xi, t)$ $Z = Y^2(0, \xi, t) = Y^2(0, \xi, t)$	$\underset{\sim}{r} = R(\xi, t)\, \underset{\sim}{i}_3 + Z(\xi, t)\, \underset{\sim}{i}_2 + Y^1\, \underset{\sim}{i}_1$ Evaluation at $Y^1 = 0$. $R = Y^3(Y^1, \xi, t) = Y^3(0, \xi, t)$ $Z = Y^2(Y^1, \xi, t) = Y^2(0, \xi, t)$	$\underset{\sim}{r} = R(\xi, t)\, \underset{\sim}{i}_3 + Z(\xi, t)\, \underset{\sim}{i}_2$ $R = Y^3(0, \xi, t)$ $Z = Y^2(0, \xi, t)$
$\underset{\sim}{a}_\alpha = \underset{\sim}{r}_{,\alpha} = \partial \underset{\sim}{r} / \partial \xi^\alpha$	$\underset{\sim}{a}_1 = R\hat{\underset{\sim}{\theta}}$, $\underset{\sim}{a}_2 = R_{,2}\, \hat{\underset{\sim}{r}} + Z_{,2}\, \hat{\underset{\sim}{k}}$	$\underset{\sim}{a}_1 = \underset{\sim}{i}_1$, $\underset{\sim}{a}_2 = R_{,2}\, \underset{\sim}{i}_3 + Z_{,2}\, \underset{\sim}{i}_2$	same
$\underset{\sim}{r}_{,\alpha\beta} = \partial^2 \underset{\sim}{r} / \partial \xi^\alpha \partial \xi^\beta$	$\underset{\sim}{r}_{,11} = -R\, \hat{\underset{\sim}{r}}$ ($\underset{\sim}{r}_{,12} = R_{,2}\, \hat{\underset{\sim}{\theta}}$ unused) $\underset{\sim}{r}_{,22} = R_{,22}\, \hat{\underset{\sim}{r}} + Z_{,22}\, \hat{\underset{\sim}{k}}$	$\underset{\sim}{r}_{,11} = \underset{\sim}{r}_{,12} = \underset{\sim}{r}_{,21} = \underset{\sim}{0}$ $\underset{\sim}{r}_{,22} = R_{,22}\, \underset{\sim}{i}_3 + Z_{,22}\, \underset{\sim}{i}_2$	same same
$a_{\alpha\beta} = \underset{\sim}{a}_\alpha \cdot \underset{\sim}{a}_\beta$	$a_{11} = (R)^2$ ($a_{12} = a_{21} = 0$) $a_{22} = (R_{,2})^2 + (Z_{,2})^2$ ($= (D)^2$)	$a_{11} = 1$ same	same same
$a = a_{11}a_{22} - (a_{12})^2$	$a = a_{11}a_{22}$ ($= (RD)^2$) ($a^{1/2} = RD$)	$a = a_{11}a_{22}$ ($= (D)^2$)	same
$a^{11} = a_{22}/a$, $a^{22} = a_{11}/a$ $a^{12} = a^{21} = -a_{12}/a$	same ($a^{11} = 1/(R)^2$, $a^{22} = 1/(D)^2$) ($a^{12} = a^{21} = 0$)	$a^{11} = 1$, $a^{22} = 1/(D)^2$ same	same same

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\underset{\sim}{n} = (\underset{\sim}{a}_1 \times \underset{\sim}{a}_2)/a^{\frac{1}{2}}$ $(a^{\frac{1}{2}} = \lvert \underset{\sim}{a}_1 \times \underset{\sim}{a}_2 \rvert)$	$\underset{\sim}{n} = n_r \hat{\underset{\sim}{r}} + n_k \hat{\underset{\sim}{k}}$	$\underset{\sim}{n} = n_r \underset{\sim}{i}_3 + n_k \underset{\sim}{i}_2$	same
	$n_r = Z_{,2}/D$, $n_k = -R_{,2}/D$	same	same
$b_{\alpha\beta} = \underset{\sim}{n} \cdot \underset{\sim}{r}_{,\alpha\beta}$	$b_{11} = -n_r R$	$b_{11} = 0$	same
	$b_{22} = n_r R_{,22} + n_k Z_{,22}$	same	same
$b^{\beta}_{\alpha} = a^{\beta\gamma} b_{\gamma\alpha}$	$b^1_1 = a^{11} b_{11}$, $b^2_2 = a^{22} b_{22}$	same $b^1_1 = 0$	same
$\underset{\sim}{a}^{\alpha} = a^{\alpha\beta} \underset{\sim}{a}_{\beta}$	$\underset{\sim}{a}^1 = a^{11} R \hat{\underset{\sim}{\theta}}$	$\underset{\sim}{a}^1 = a^{11} \underset{\sim}{i}_1$	same
	$\underset{\sim}{a}^2 = a^{22} R_{,2} \hat{\underset{\sim}{r}} + a^{22} Z_{,2} \hat{\underset{\sim}{k}}$	$\underset{\sim}{a}^2 = a^{22} R_{,2} \underset{\sim}{i}_3 + a^{22} Z_{,2} \underset{\sim}{i}_2$	same
$\Gamma_{\alpha\beta}{}^{\gamma} = \underset{\sim}{a}^{\gamma} \cdot \underset{\sim}{r}_{,\alpha\beta}$	$\Gamma_{11}{}^2 = -a^{22} R_{,2} R$	$\Gamma_{11}{}^2 = 0$	same
	$\Gamma_{22}{}^2 = a^{22}(R_{,2} R_{,22} + Z_{,2} Z_{,22})$	same	same
	$\{\Gamma_{12}{}^1 = \Gamma_{21}{}^1 = a^{11} R_{,2} R$ unused$\}$	$\{\Gamma_{12}{}^1 = \Gamma_{21}{}^1 = 0\}$	same
$\Delta\underset{\sim}{u} = (\underset{\sim}{r} - \underset{\sim}{r}^-)$ $\{= \Delta u^k \underset{\sim}{i}_k = \Delta\underset{\sim}{r}\}$	$\Delta\underset{\sim}{r} = \Delta R \hat{\underset{\sim}{r}} + \Delta Z \hat{\underset{\sim}{k}}$	$\Delta\underset{\sim}{r} = \Delta R \underset{\sim}{i}_3 + \Delta Z \underset{\sim}{i}_2$	same
$\Delta n_{\alpha} = -\underset{\sim}{n}^- \cdot \Delta\underset{\sim}{u}_{,\alpha}$	$\Delta n_2 = -(n_r^- \Delta R_{,2} + n_k^- \Delta Z_{,2})$	same	same
$\Delta\underset{\sim}{n} = (\underset{\sim}{a}_{\alpha} + \frac{\Delta n_{\alpha}}{1 + \underset{\sim}{n}^- \cdot \underset{\sim}{n}} \underset{\sim}{n}) a^{\alpha\beta} \Delta n_{\beta}$	$\Delta\underset{\sim}{n} = \Delta n_r \hat{\underset{\sim}{r}} + \Delta n_k \hat{\underset{\sim}{k}}$	$\Delta\underset{\sim}{n} = \Delta n_r \hat{\underset{\sim}{r}} + \Delta n_k \hat{\underset{\sim}{k}}$	same
	$\Delta n_r = (R_{,2} + n_r dn) a^{22} \Delta n_2$	same	same
	$\Delta n_k = (Z_{,2} + n_k dn) a^{22} \Delta n_2$	same	same
	$dn = \Delta n_2/(1 + n_r n_r^- + n_k n_k^-)$	same	same

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$g_{\alpha\beta} = a_{\alpha\beta} - 2\zeta b_{\alpha\beta}$	same $(g_{12} = g_{21} = 0)$	same $(g_{11} = 1)$	same (g_{11} not used)
$g = g_{11}g_{22} - (g_{12})^2$	$g = g_{11}g_{22}$	same $(g = g_{22})$	same
$g^{11} = g_{22}/g$, $g^{22} = g_{11}/g$	same $(g^{11} = 1/g_{11}$, $g^{22} = 1/g_{22})$	same $(g^{11} = 1)$	same (g^{11} not used)
$g^{12} = g^{21} = -g_{12}/g$	$g^{12} = g^{21} = 0$	same	same
$\Delta\epsilon_{\alpha\beta} = \frac{1}{2}(\underset{\sim}{a}_\alpha \cdot \Delta\underset{\sim}{u}_{,\beta} + \underset{\sim}{a}_\beta \cdot \Delta\underset{\sim}{u}_{,\alpha}$	$\Delta\epsilon_{\alpha\beta} = \frac{1}{2}\Delta a_{\alpha\beta} - \zeta\Delta b_{\alpha\beta}$ $(\Delta\epsilon_{12} = \Delta\epsilon_{21} = 0)$	same $(\Delta\epsilon_{11} = 0)$	same
$- \Delta\underset{\sim}{u}_{,\alpha} \cdot \Delta\underset{\sim}{u}_{,\beta})$	$\frac{1}{2}\Delta a_{11} = \Delta R(R_{,2} - \frac{1}{2}\Delta R_{,2})$	$\Delta a_{11} = 0$	same
$- \zeta(\underset{\sim}{n}^- \cdot \Delta\underset{\sim}{u}_{,\alpha\beta} + \Delta\underset{\sim}{n} \cdot \underset{\sim}{r}_{,\alpha\beta})$	$\frac{1}{2}\Delta a_{22} = \Delta R_{,2}(R_{,2} - \frac{1}{2}\Delta R_{,2}) + \Delta Z_{,2}(Z_{,2} - \frac{1}{2}\Delta R_{,2})$	same	same
	$\Delta b_{11} = -(\Delta R n_r^- + R\Delta n_r)$	$\Delta b_{11} = 0$	same
	$\Delta b_{22} = (n_r^-\Delta R_{,22} + n_k^-\Delta Z_{,22}) + (\Delta n_r R_{,22} + \Delta n_k Z_{,22})$	same	same
$\Delta\epsilon_\beta^\alpha = g^{\alpha\gamma}\Delta\epsilon_{\gamma\beta}$	$\Delta\epsilon_1^1 = g^{11}\Delta\epsilon_{11}$, $\Delta\epsilon_2^2 = g^{22}\Delta\epsilon_{22}$	same $(\Delta\epsilon_1^1 = 0)$	same
$\sigma^{-\alpha}_{\ \beta} = g_{\beta\gamma}\sigma^{-\gamma\alpha}$	$\sigma^{-1}_{\ 1} = g_{11}\sigma^{-11}$, $\sigma^{-2}_{\ 2} = g_{22}\sigma^{-22}$	same	same ($\sigma^{-1}_{\ 1}$ not used)
$\Delta\sigma^{E_\alpha}_\beta = \frac{E}{1+\nu}(\Delta\epsilon^\alpha_\beta + \frac{\nu}{1-\nu}\Delta\epsilon^\gamma_\gamma\delta^\alpha_\beta)$	$\Delta\sigma^{E_1}_1 = (\Delta\epsilon^1_1 + \nu\Delta\epsilon^2_2)E/(1-\nu^2)$	same	same ($\Delta\sigma^{E_1}_1$ not used)
	$\Delta\sigma^{E_2}_2 = (\nu\Delta\epsilon^1_1 + \Delta\epsilon^2_2)E/(1-\nu^2)$	same	same
$\sigma^{T_\alpha}_\beta = \sigma^{-\alpha}_{\ \beta} + \Delta\sigma^{E_\alpha}_\beta$	$\sigma^{T_1}_1 = \sigma^{-1}_{\ 1} + \Delta\sigma^{E_1}_1$, $\sigma^{T_2}_2 = \sigma^{-2}_{\ 2} = \Delta\sigma^{E_2}_2$	same	same ($\sigma^{T_1}_1$ not used)

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\psi_T = \frac{1}{2}[3\sigma^{T}_{\beta}\sigma^{T}_{\alpha} - (\sigma^{T}_{\gamma})^2 - (\sigma_0)^2$	$\psi_T = \sigma^{T_1}_{1}(\sigma^{T_1}_{1} - \sigma^{T_2}_{2}) + (\sigma^{T_2}_{2})^2 - (\sigma_0)^2$	same	$(\psi_T = (\sigma^{T_2}_{2})^2 - (\sigma_0)^2)$
If $\psi_T < 0$, $\sigma^{\alpha}_{\beta} = \sigma^{T_\alpha}_{\beta}$	Same for σ^1_1 and σ^2_2	same	
If $\psi_T > 0$, the stress increment is partly plastic. Correction is	same	same	If $\lvert\sigma^{T_2}_{2}\rvert \leq \sigma_0$, $\sigma^2_2 = \sigma^{T_2}_{2}$ The uniaxial stress-strain assumption gives the simple corrections:
$\sigma^{C_\alpha}_{\beta} = \sigma^{-\alpha}_{\beta} - \frac{1-2\nu}{3(1-\nu)}\sigma^{-\gamma}_{\gamma}\delta^{\alpha}_{\beta}$	$\sigma^{C_{-1}}_{1} = (2-\nu)\sigma^{-1}_{1} - (1-2\nu)\sigma^{-2}_{2}$	same	If $\sigma^{T_2}_{2} > \sigma_0$, $\sigma^2_2 = \sigma_0$
	$\sigma^{C_{-2}}_{2} = (2-\nu)\sigma^{-2}_{2} - (\frac{1-2\nu}{1-\nu})\sigma^{-1}_{1}$	same	If $\sigma^{T_2}_{2} < -\sigma_0$, $\sigma^2_2 = -\sigma_0$
$\sigma^{\alpha}_{\beta} = \sigma^{I_\alpha}_{\beta} - \lambda\sigma^{C_\alpha}_{\beta}$	same for σ^1_1 and σ^2_2, where	same	
where λ is the smaller root of	$A\lambda^2 - B\lambda + C = 0$	same	
$3\sigma^{\alpha}_{\beta}\sigma^{\beta}_{\alpha} - (\sigma^{\gamma}_{\gamma})^2 = 2(\sigma_0)^2$	$A = \sigma^{C_1}_{1}(\sigma^{C_1}_{1} - \sigma^{C_2}_{2}) + (\sigma^{C_2}_{2})^2$, $C = \psi_T$	same	
	$B = \sigma^{T_1}_{1}(2\sigma^{C_1}_{1} - \sigma^{C_2}_{2}) + \sigma^{T_2}_{2}(2\sigma^{C_2}_{2} - \sigma^{C_1}_{1})$	same	
$\sigma^{\alpha\beta} = g^{\alpha\gamma}\sigma^{\beta}_{\gamma}$	$\sigma^{11} = g^{11}\sigma^1_1$, $\sigma^{22} = g^{22}\sigma^2_2$	same	$\sigma^2_2 = g^{22}\sigma^2_2$
This stress calculation is for an elastic perfectly plastic material with no complications. The mechanical sublayer model is used for strain hardening. Strain rate dependency may be used with the equations:		same	same
$\sigma_0 = \sigma_{0(\text{static})}[1 + (\dot{\epsilon}/d)^{(1/p)}]$	same	same	same
$\dot{\epsilon} = \frac{1}{\Delta t}[\frac{3}{2}\Delta\epsilon^{\alpha}_{\beta}\Delta\epsilon^{\beta}_{\alpha} - \frac{1}{2}(\Delta\epsilon^{\gamma}_{\gamma})^2]^{\frac{1}{2}}$	$\dot{\epsilon} = [\Delta\epsilon^1_1(\Delta\epsilon^1_1 - \Delta\epsilon^2_2) + (\Delta\epsilon^2_2)^2]^{\frac{1}{2}}/\Delta t$	same	$\dot{\epsilon} = \Delta\epsilon^2_2/\Delta t$
If ψ_T is large, the above procedure is repeated L times with $\Delta\sigma^{E_\alpha}_{\beta}/L$ in place of $\Delta\sigma^{E_\alpha}_{\beta}$. If λ is negative or complex, L is increased. If L is too large, or if $g \leq 0$, the run is aborted.		same	No iteration.

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\sigma^{\alpha\beta}$ for each pseudo-sublayer is stored for future use. A weighted stress, $\sigma_k^{\alpha\beta}$ is found for layer k.	same for σ^{11} and σ^{22}	same	same for σ^{22}
$\Sigma_\alpha^{\alpha\beta} = \sum_k \sigma_k^{\alpha\beta} (\zeta_k)^\alpha \Delta\zeta$, $(\alpha=0,1,2)$	same for Σ_α^{11} and Σ_α^{22}	same	$\Sigma_\alpha^{22} = \sum_k W_k \sigma_k^{22} (\zeta_k)^\alpha$ $\{W_k = \text{Area}_k\}$
$\hat{Q}^{*\alpha\beta} = a^{-\frac{1}{2}}(\Sigma_0^{\alpha\beta} - b_\gamma^\gamma \Sigma_1^{\alpha\beta})$	$\hat{Q}^{*11} = a^{-\frac{1}{2}}[\Sigma_0^{11} - (b_1^1 + b_2^2)\Sigma_1^{11}]$	same $\{b_1^1 = 0\}$	$Q^{*11} = 0$ not used
	$\hat{Q}^{*22} = a^{-\frac{1}{2}}[\Sigma_0^{22} - (b_1^1 + b_2^2)\Sigma_1^{22}]$	same	same
$\hat{M}^{*\alpha\beta} = a^{-\frac{1}{2}}[\Sigma_1^{\alpha\beta} - b_\gamma^\gamma \Sigma_2^{\alpha\beta} - b_\gamma^{(\alpha}\Sigma_2^{\gamma\beta)}]$	$\hat{M}^{*11} = a^{-\frac{1}{2}}[\Sigma_1^{11} - (2b_1^1 + b_2^2)\Sigma_2^{11}]$	same	$M^{*11} = 0$ not used
	$\hat{M}^{*22} = a^{-\frac{1}{2}}[\Sigma_1^{22} - (b_1^1 + 2b_2^2)\Sigma_2^{22}]$	same	same
$\underset{\sim}{\hat{N}}^{*\alpha} = \hat{Q}^{*\alpha\beta}\,\underset{\sim}{a}_\beta + \Gamma_{\beta\gamma}{}^\alpha M^{*\beta\gamma}\,\underset{\sim}{n}$	$\underset{\sim}{\hat{N}}^{*1} = FN_\theta^1\,\underset{\sim}{\hat{\theta}} = Q^{*11}R\,\underset{\sim}{\hat{\theta}}$	$\underset{\sim}{\hat{N}}^{*1} = \hat{Q}^{*11}\,\underset{\sim}{i}_1$ not used	same $\{\underset{\sim}{N}^{*1} = 0\}$
	$\underset{\sim}{\hat{N}}^{*2} = FN_r^2\,\underset{\sim}{\hat{r}} + FN_k^2\,\underset{\sim}{\hat{k}}$	$\underset{\sim}{\hat{N}}^{*2} = FN_r^2\,\underset{\sim}{i}_3 + FN_k^2\,\underset{\sim}{i}_2$	same
	$FN_r^2 = \hat{Q}^{*22}R_{,2} + n_r CSM^2$	same	same
	$FN_k^2 = \hat{Q}^{*22}Z_{,2} + n_k CSM^2$	same	same
	$CSM^2 = \Gamma_{11}{}^2\,\hat{M}^{*11} + \Gamma_{22}{}^2\,\hat{M}^{*22}$	same $\{\Gamma_{11}{}^2 = 0\}$	same
$\underset{\sim}{\hat{M}}^{*\alpha\beta} = \hat{M}^{*\alpha\beta}\,\underset{\sim}{n}$	$\underset{\sim}{\hat{M}}^{*\alpha\alpha} = FM_r^{\alpha\alpha}\,\underset{\sim}{\hat{r}} + FM_k^{\alpha\alpha}\,\underset{\sim}{\hat{k}}$ (no sum)	$\underset{\sim}{\hat{M}}^{*\alpha\alpha} = FM_r^{\alpha\alpha}\,\underset{\sim}{i}_3 + FM_k^{\alpha\alpha}\,\underset{\sim}{i}_2$ (no sum)	same $(\underset{\sim}{M}^{*11} = 0)$
	$FM_r^{11} = \hat{M}^{*11}n_r$, $FM_r^{22} = \hat{M}^{*22}n_r$	same	same
	$FM_k^{22} = \hat{M}^{*22}n_k$ $\{FM_k^{11}$ not used$\}$	same	same

Standard REPSIL	Axial Symmetry	Slab Symmetry	Laterally Symmetric Beam
$\underset{\sim}{F}^* = -a^{1/2}\Delta P\ \underset{\sim}{n}$	$\underset{\sim}{F}^* = F_r^*\ \hat{r} + F_k^*\ \hat{k}$	$\underset{\sim}{F}^* = F_r^*\ \underset{\sim}{i}_3 = F_k^*\ \underset{\sim}{i}_2$	same
	$F_r^* = -a^{1/2}\Delta P n_r\ ,\ F_k^* = -a^{1/2}\Delta P n_k$	same	same
$\Delta\underset{\sim}{u}^+ = \Delta\underset{\sim}{u} + \Delta t^2(\underset{\sim}{\hat{M}}^{*\alpha\beta}{}_{,\alpha\beta} + \underset{\sim}{\hat{N}}^{*\alpha}{}_{,\alpha}$	$\Delta R^+ = \Delta R + \Delta t^2(FM_r^{22}{}_{,22} - FM_r^{11}$	$\Delta R^+ = \Delta R + \Delta t^2(FM_r^{22}{}_{,22} + FN_r^2{}_{,2}$	same
$+ \underset{\sim}{F}^*)/\gamma_0^*$	$+ FN_r^2{}_{,2} - FN_\theta^1 + F_r^*)/\gamma_0^*$	$+ F_r^*)/\gamma_0^*$	
$= \Delta\gamma^{k+}\ \underset{\sim}{i}_k$	$\Delta Z^+ = \Delta Z + \Delta t^2(FM_k^{22}{}_{,22} + FN_k^2{}_{,2}$	same	same
	$+ F_k^*)/\gamma_0^*$		
$\gamma_0^* = \rho h A^{1/2}$ (A = a at time t=0)	same	same	$\gamma_0^* = \rho A^{1/2} \sum_k W_k$
$\underset{\sim}{r}^+ = \underset{\sim}{r} + \Delta\underset{\sim}{u}^+$	same	same	same
$\gamma^{k+} = \gamma^k + \gamma^{k+}$	$R^+ = R + \Delta R^+\ ,\ Z^+ = Z + \Delta Z^+$	same	same

APPENDIX B

DEFINITION OF FORTRAN VARIABLES

In this appendix we list the FORTRAN variables used in RPSL1D as listed in Appendix C, plus additional variables used in the options PLOTP, SHR3/1, MSQSVS, APLFRC, EAPFRC, EROD, or BSTRS. Extra columns and a lot of symbology were used in an attempt to shorten the appendix but still give enough information.

Column 1 is the FORTRAN name. Closely related names may be grouped. Arrays are indicated by enclosing a symbolic integer in parentheses (e.g. AMS(N)).

Column 2 lists the symbol used in this text or one of the references. If the symbol is in braces, { }, it was first used in this report; if in brackets, [ ], it was first used in Reference 2; if in neither braces or brackets, the symbol originated in Reference 1. Examples:

{M} Symbol for moment in this report.

[A] Coefficient used in Reference 2.

a_{11} Symbol of a covariant metric component first used in Reference 1.

Column 3 partially describes the status of the variable:

C denotes a name in COMMON.

I denotes input.

S_0 denotes a variable that is rarely changed throughout a computer run, as opposed to

S denoting a term that is fixed for one time step, or

T denoting a transient value.

D denotes a dummy argument for a subroutine.

Column 4 names the subroutine which stores the variable. (If other subroutines change or store these variables, they will usually be listed in Column 6.)

Column 5 gives the stage of development when the variable originated:

U denotes a variable described in Appendix C of the User's Manual (Reference 2),

R denotes a newer name included in the RPSL1D file,

any name is the option that first used the variable.

Column 6 defines the variable and may give additional information such as a reference, a limiting value, a formula, the input card that it is entered on, or any other item of interest.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
AA	[A]	T	STRESS	U	Coefficient for computing TAMBDA.
AMS(N)	[M]	C,S	RESULT	SUR3/1	Moment at mesh point N. AMS = A22*F22.
ALFN		T	STRAIN	U	Alphanumeric for output: "INNER" or "OUTER".
ANGEL	[θ]	T	STRAIN	U	ANGLE(I) or ANGLB(I) in radians.
ANGLB(I), ANGLE(I)	[θ]	I,C,S_0	START	U	Angles to locate surface strains. (Input 13 in degrees) (Ref. 2, p. 52)
AR11, AR22	a^{11},a^{22}	T	DGEOM	U	Contravariant components, reference surface metric.
ASA(I), ASB(I)	$[2\alpha^2]$	C,S	STRAIN	U	(α = sinθ. Ref. 2, p. 52)
AY3MN	$\underline{u} \cdot \underline{u}$	T	STRAIN	R	Square of displacement.
A11,A22	a_{11},a_{22}	C,T	DGEOM	U	Covariant components, reference surface metric.
A111(I),A221(I)	A_{11},A_{22}	C,S_0	ABINIT	U	Initial a_{11},a_{22} at NI1(I).
A112(I),A222(I)	A_{11},A_{22}	C,S_0	ABINIT	U	Initial a_{11},a_{22} at NI2(I).
B	[-2B]	T	STRESS	U	Coefficient for computing TAMBDA.
BEPL,BEPLM,BEPS		T	BMSTRS	BSTRS	Temporary.
BM11,BM22	b_1^1,b_2^2	C,T	DGEOM	U	Mixed components of 2nd fundamental tensor.
BSA(I),BSB(I)	$[2\beta^2]$	C,S_0	STRAIN	U	(β = cosθ. Ref. 2, P. 52)
BT	$b_1^1+b_2^2$	C,T	DGEOM	U	Trace of 2nd fundamental tensor.
B11,B22	b_{11},b_{22}	C,T	DGEOM	U	Covariant components, 2nd fundamental tensor.
B111(I),B221(I)	B_{11},B_{22}	C,S_0	ABINIT	U	Initial b_{11},b_{22} at NI1(I).
B112(I),B222(I)	B_{11},B_{22}	C,S_0	ABINIT	U	Initial b_{11},b_{22} at NI2(I).
C	[c] (C)	T	START	U	Sound speed. $C=\sqrt{E/(\rho(1-\nu^2))}$. If IB>0, $C=\sqrt{E/\rho}$.
CA	$\Delta\xi^1 * \Delta\xi^2$	C,S_0	START	U	Surface area. Used for kinetic energy and work.
CB		C,S_0	START	U R R	CB = $\frac{1}{2}\Delta\zeta\Delta\xi^1\Delta\xi^2/E$. If IB>0, CB = $\frac{1}{2}\Delta\xi^2/E$. If IGAUSS = 1, CB = $\frac{1}{4}h\Delta\xi^1\Delta\xi^2/E$.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
CINEP	$[T^*]$	C,S	DAMP	U	Kinetic energy, KE, removed by KEA. (Ref. 2, p.46)
CINER	$[T_-]$	C,S	KINET	U	KE at time t-Δt/2.
CINES	$[T_+, T_{++}]$	C,S	KINET	U	KE at time t+Δt/2 or t+3Δt/2. (REPSLID,DAMP,DESTEP)
CINES1	$[\ ]$	C,T	DAMP	U	Previous CINER in DAMP and DESTEP.
CINES2		C,T	DAMP	R	Previous CINES in DAMP. (Unused)
CINET	[T]	C,S	KINET	U	Kinetic energy at current time, t. (DESTEP)
CN		T	KINET	U	Portion of CA used with mesh point. (PWORK)
CSM2		T	RESULT	U	Normal component of $\underset{\sim}{N}^{*2}/TB$.
CS211,CS222	$\Gamma_{11}^2, \Gamma_{22}^2$	C,T	DGEOM	U	Christoffel symbols. Used in RESULT.
CTWO	$\{C_2\}$	T	START	R	Constant for computing DELM. (See 3.5)
C1	[C2/(C2+4)]	C,S_0	START	U	Constant for equation of motion. (DESTEP)
C1OLD		T	DESTEP	U	Old value of C1 in DESTEP.
C2		C,S_0	START	U	(C2 = 2 * DELTAT * DAMPF / GAMZ.) (DESTEP)
D	$(a_{22})^{\frac{1}{2}}$	T	DGEOM	R	Temporary storage in DGEOM, MOTION, and INNORM.
D	$[B^2-AC]$	T	STRESS	U	Discriminant of quadratic equation for TAMBDA.
DA	a	T	DGEOM	U	Determinant of surface metric. DA = A11*A22.
DAMPF	[D]	C,I,S_0	START	U	Viscous damping coefficient. (Input 5)
DAT(I)		T	PDATA	U	Array for storage on plotting file.
DA1(N)	$\frac{1}{2}(a_{11}-A_{11})$	C,S	DGEOM	R	Σ DA11 at mesh point N. (A11 at time 0.)
DA2(N)	$\frac{1}{2}(a_{22}-A_{22})$	C,S	DGEOM	R	Σ DA22 at mesh point N. (A22 at time 0.)
DA11,DA22	$\frac{1}{2}\Delta a_{11}, \frac{1}{2}\Delta a_{22}$	T	DGEOM	U	Half the increment in a_{11}, a_{22} from t-Δt to t.
DA22P	$\frac{1}{2}\Delta a_{22}$	T	DGEOM	R	$\frac{1}{2}\Delta a_{22}$ at previous midmesh.
DB1(N)	$b_{11}-B_{11}$	C,S	DGEOM	R	Σ DB11 at mesh point N.
DB2(N)	$b_{22}-B_{22}$	C,S	DGEOM	R	Σ DB22 at mesh point N.
DB11,DB22	$\Delta b_{11}, \Delta b_{22}$	T	DGEOM	U	Increment in b_{11}, b_{22} from t-Δt to t.
DE		C,S	ERODE	EROD	Erosion from end mesh point. May be negative.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
DECOM	$(\Delta\varepsilon/4)$	S_0	ERODE	EROD	Maximum permitted DE before changing end point.
DEFX		C,S	STRAIN	R	Maximum displacement.
DEFXT		T	STRAIN	R	Maximum displacement in current cycle.
DELB	$[\Delta t_B]$	T	START	U R R	Critical bending time increment. $\Delta t_B = \frac{1}{2}(DETA2)^2/(C\sqrt{CTWO})$ $\Delta t_B = \sqrt{3}(DETA2)^2/(Ch)$, if IGAUSS = 1.
DELDE		T	ERODE	EROD	Erosion increment from t-Δt to t.
DELGAM	$(\Delta t)^2/\gamma_0)$ $(\Delta t)^2/GAMZ$	C,S_0	RPSLID	U	Constant used in DGEOM for TEMP(N).
DELIN	Δt	T,I	START	U	Temporary storage for input Δt.
DELM	$[\Delta t_M]$	T	START	U	Critical membrane time increment. $\Delta t_M = \Delta\varepsilon/C$.
DELMIN		T	START	U	0.95*MIN[DELB,DELM] rounded.
DELR,DELS		T	DESTEP	U	Factors used in DESTEP to change DELTAT.
DELSQ	$(\Delta t)^2$	C,S_0	START	U	Square of time increment. (DESTEP)
DELTAT	Δt	C,I,S_0	START	U	Time increment. (Input 13) May change to minimum of DELIN and DELMIN. (DESTEP)
DELTR,DELTZ		T	BOUNDR	R	Components used to locate external point.
DEPS1(K), DEPS2(K)	$\Delta\varepsilon_{11}$,$\Delta\varepsilon_{22}$	C C	DGEOM DGEOM	U U	Covariant components of strain increment at ζ_k. $\Delta\varepsilon_{\alpha\beta} = \frac{1}{2}\Delta a_{\alpha\beta} - \zeta_k \Delta b_{\alpha\beta}$.
DEPS11,DEPS22	$\Delta\varepsilon^1_1$,$\Delta\varepsilon^2_2$	T	STRESS	U	Mixed components of strain increment. (BMSTRS)
DETA1	$\Delta\xi^1$ $[\Delta n^1]$	C,S_0	INGEOM	U	Increment in the ξ^1,ξ^2 Lagrange coordinates.
DETA2	$\Delta\xi^2$ $[\Delta n^2]$ $\{\Delta\xi\}$	C,S_0	INGEOM	U	(If IB=0, $\Delta\xi^1$=1.0 in START.)
DFACT		C,I,S_0	START	U	Factor for terminating damped run. (Input 5)
DFDG		S_0	PRESS		In PRESS of 10/12/76, DFDG = -DAMPF/GAMZ.
DG	g	T	STRESS	U	Determinant of metric. DG = G11*G22.
DN	Δn	T	DGEOM	U	Normal component of $\Delta\underline{n}$. $\Delta\underline{n} = \Delta n_2\,\underline{g}^2 + \Delta n\,\underline{n}$.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
DN1(I),DN2(I),		C,S_0	START	U	Interpolation coefficients for surface strains.
DN3(I)		C,S_0	START	R	DN3(I) added with revision.
DNK,DNR	$\{\Delta n_k, \Delta n_r\}$	T	DGEOM	R	Cartesian components of $\Delta \underline{n}$. $\Delta \underline{n}$ = DNK $\underline{i}_2$ + DNR $\underline{i}_3$.
DNL2,DNR2	$\Delta n_2, \Delta n^2$	T	DGEOM	U	Covariant, contravariant tangential component of $\Delta \underline{n}$.
DR(N),DZ(N)	$\{\dot{R}, \dot{Z}\}$ $\{\Delta R, \Delta Z\}$	C,S	INVEL	R	[Replaces $U_j(M,N)$] Components of velocity (INVEL). Displacement increments RPSLID, MOTION, and other.
DRI,DZI	$\{\Delta R^-, \Delta Z^-\}$	T	MOTION	R	Components of displacement increment, t-Δt to t.
DRI1,DZI1	$\{\Delta R^-, \Delta Z^-\}$	T	MOTION	EAPFRC	DRI and DZI at end 1.
DRMSN	$\{\Delta R\}$	T	DGEOM	R	Radial displacement increment.
DRS,DZS	$\{\Delta R, \Delta Z\}$	T	MOTION	R	Components of displacement increment, t to t+Δt.
DR2,	$\{\Delta R,_2\}$	C,T	DGEOM	R	[Replace U2i.] First finite differences of displacement increment components.
DZ2	$\{\Delta Z,_2\}$	C,T	DGEOM	R	
DR22,	$\{\Delta R,_{22}\}$	C,T	DGEOM	R	[Replace U22i.] Second finite differences of displacement increment components.
DZ22	$\{\Delta Z,_{22}\}$	C,T	DGEOM	R	
DSG11L,DSG22L	$\Delta\sigma^{E\alpha}_\beta/L$	T	STRESS	U	Subincrement division of $\Delta\sigma^{E\alpha}_\beta$.
DS1G11,DS1G22	$\Delta\sigma^{E1}_1, \Delta\sigma^{E1}_1$	T	STRESS	U	Mixed components of elastic stress increment.
DSQOLD		T	DESTEP	U	Old $(\Delta t)^2$.
DSR(J)	$\{d_j\}$	C,I,S_0	START	U	Constant for strain rate (Input 7).
DTHET	$\{\Delta\theta\}$	T	MOTION	EAPFRC	Angle increment [t,t+Δt] current end.
DTH1P,DTH2P	$\{\Delta\theta^-_j\}$	C,S_0	MOTION	EAPFRC	Angle increment [t-Δt,t] (end 1,end 2).
DWE1P,DWE2P	$\{\Delta W^-_j\}$	C,S_0	MOTION	EAPFRC	Work increment [t-Δt,t] (end 1, end 2).
DWF	$\{\Delta W_F\}$	T	MOTION	EAPFRC	Work increment [t,t+Δt], from force on the end.
DWM	$\{\Delta W_M\}$	T	MOTION	EAPFRC	Work increment [t,t+Δt], from moment on the end.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
DW	$\underline{n} \cdot \Delta\underline{r}$	T	PWORK	U	Normal component of $\Delta\underline{r}$.
DZ(N)	$\{\Delta Z\}$	C,S	MOTION	R	See DR(N).
DZETAK	$\{\Delta\zeta_k\}$	I,T	INGEOM	R	Thickness of k'th layer of a beam. (Input 14)
DZI, DZII, DZS		T	MOTION	R	See DRI, DRII, DRS.
DZ2,DZ22		C,T	DGEOM	R	See DR2 and DR22.
D1,D2	$[U^3,U^2]$	C,S	STRAIN	U	Components of displacement at ETAD2. $\{\Delta R,\Delta Z\}$
E	E	C,I,S_0	START	U	Young's modulus. (Input 6)
EEB(I),SSB(I)	$\{\epsilon_i,\sigma_i\}$	C,I,S_0	STAR	BSTRS	Uniaxial stress-strain curve. 1<I<100. (Input 7) (See comments at SEPS(J).)
EFMU	$\{\mu(t)\}$	T	ENDFRC	APLFRC	μ at time t. EFR1 = EFMU * EFZ1.
EFMUT(I),TMFRI(I)	$\{\mu_i,t_i\}$	I,S_0	ENDFRC	APLFRC	Table of μ vs. time. (Input 18)
EFR1		C,S	ENDFRC	APLFRC	$\underline{i}_3$ force, end 1. EFR1>0 decreases ΔR(N1B).
EFR2		C,S	ENDFRC	APLFRC	$\underline{i}_3$ force, end 2. EFR2>0 increases ΔR(N2B).
EFTH	$\{\theta(t)\}$	T	ENDFRC	APLFRC	θ at time t.
EFTHT(I),TMTH(I)	$\{\theta_i,t_i\}$	I,S_0	ENDFRC	APLFRC	Table of θ vs. time. (Input 18)
EFZT(I),TMFZ(I)		I,S_0	ENDFRC	APLFRC	Table of force, EFZ1, vs. time, t. (Input 18)
EFZ1		C,S	ENDFRC	APLFRC	$\underline{i}_2$ force, end 1. EFZ1>0 decreases ΔZ(N1B).
EFZ2		C,S	ENDFRC	APLFRC	$\underline{i}_2$ force, end 1. EFZ2>0 increases ΔZ(N2B).
EM1		C,S	ENDFRC	APLFRC	Applied moment at end 1. EM1>0 decreases ΔR(N1B), increases ΔR(N1B+1).
EM2		C,S	ENDFRC	APLFRC	Applied moment at end 2. EM2>0 increases ΔR(N2B), decreases ΔR(N2B-1).
EN	$[\Delta W(t-\frac{1}{2}\Delta t)]$	T	MOTION	U	Work increment, $[t-\Delta t,t]$. (EN = $\frac{1}{2}$CA*(EMR+ENS))
ENR	$[\Delta W(t-\Delta t)/CA]$	C,S	MOTION	U	Work increment/CA, $[t-\frac{3}{2}\Delta t,t-\frac{1}{2}\Delta t]$. (RPSL10)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
ENS	$[\Delta W(t)/CA]$	C,S	PWORK	U	Work increment/CA, $[t-\frac{1}{2}\Delta t, t+\frac{1}{2}\Delta t]$. (RPSLID)
EPC(I)		T	STRAIN	R	Temporary for computing extreme surface strains.
EPCL1,EPCL2	$[E_1, E_2]$	T	DGEOM	R	Elongational components of strain on the lower surface in ξ^1 and ξ^2 directions. (Output)
EPCU1,EPCU2	$[E_1, E_2]$	T	DGEOM	R	Elongational components of strain on the upper surface in ξ^1 and ξ^2 directions. (Output)
EPLB(N,K)	$\{\epsilon_m\}$	C,S	BMSTRS	BSTRS	Mean of possible elastic variation of EPSB(N,K).
EPLBM(N,K)	$\{\epsilon_m\}$	C,S	BMSTRS	BSTRS	Mean of possible elastic variation of EPSBM(N,K).
EPSANB(I) EPSANG(I)	$[E_\theta]$ $[E_\theta]$	T T	STRAIN STRAIN	U U	Elongational surface strains in θ directions at location I. (Output)
EPSB(N,K)	ϵ_2^2	C,S	BMSTRS	BSTRS	Uniaxial strain at mesh point N, layer K.
EPSBM(N,K)	ϵ_2^2	C,S	BMSTRS	BSTRS	Uniaxial strain at midmesh N, layer K.
EPSDOT	$\dot{\epsilon}$	T	STRESS	U	Deviator strain rate in strain rate equation.
EPSL1(N), EPSL2(N)	$\epsilon_{11}, \epsilon_{22}$	C	DGEOM	U	Covariant strain components on the lower surface at mesh point N.
EPSR1,EPSR2	$[\epsilon_1, \epsilon_2]$	T	STRAIN	U	Intermediate values in surface strain calculations.
EPSS1(I), EPSS2(I)	$[E_1, E_2]$	C	STRAIN	U	Elongational components of strain in the ξ^1 and ξ^2 directions at position I.
EPSU1(N), EPSU2(N)	$\epsilon_{11}, \epsilon_{22}$	C	DGEOM	U	Covariant strain components on the upper surface at mesh point N.
EPSX(I)		C,S	STRAIN	R	Extreme strains. (I = 1,8) (Output)
EPSXT(I)		T	STRAIN	R	Extreme strains at current time. (Output)
EPSZ	σ_0/E	C,S_0	START	BSTRS	Strain at yield stress.
EROD		T	ERODE	EROD	Erosion from initial end 1 up to time t.
ERODP		S	ERODE	EROD	Erosion from initial end 1 up to time t-Δt.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
ETAD2	ξ^2 {ξ} [η^2]	C,I,S_0	START	U	Material coordinate of the point at which displacement is plotted. (Input 12) (DGEOM?)
ETAG2(I)	ξ^2 {ξ} [η^2]	C,I,S_0	START	U	Material coordinate for the I'th surface strain. (Input 13) (DGEOM?)
FLOATL	[1/L]	T	STRESS	U	Proportion of stress increment per step.
FM11R(N)	$\{\hat{M}^{*11}\ n_r\}$				
FM22K(N)	$\{\hat{M}^{*22}\ n_k\}$	C,T	RESULT	R	Components of bending resultant tensor times the normal at mesh point N. (These replace $FM_{\alpha\beta}(M,N)$ of Reference 2.)
FM22R(N)	$\{\hat{M}^{*22}\ n_r\}$				
FNU		C,I,S_0	START	U	Poisson's ratio. (Input 6)
FN11H(N),FN2K(N) FN2R(N)	ν	C,T	RESULT	R	Components of stress resultant tensor at midmesh N. [These replace $FN_{\alpha j}$ of Reference 2.]
F11,F22	$\{\frac{\hat{M}^{*11}}{IB}, \frac{\hat{M}^{*22}}{IB}\}$	T	RESULT	U	Proportional to bending resultant components.
G	$\frac{1}{2}E/(1+\nu)$	C,S_0	START	U	Shear modulus. Used in STRESS.
GAMZ	γ_0^* [Γ_0]	C,S_0	START	U R	If IB=0, Γ_0 = ρh = mass/unit surface area. If IB=1, Γ_0 = ρ*SUMAR = mass/unit length.
GI11(I),GI22(I)	G^{11}, G^{22}	C,S_0	STRAIN	U	Contravariant components of initial metric at I.
GL11(N),GL22(N)	G_{11}, G_{22}	C,S_0	RPSLID	R	Initial metric, lower surface, mesh point N.
GR11,GR22	g^{11}, g^{22}	T	STRESS	U	Contravariant metric components. (BMSTRS)
GU11(N),GU22(N)	G_{11}, G_{22}	C,S_0	RPSLID	R	Initial metric, upper surface, mesh point N.
G11,G22	g_{11}, g_{22}	T	STRESS	U	Metric at ζ_k in STRESS and BMSTRS.
	G_{11}, G_{22}	T	STRAIN	U	Initial metric on surfaces in STRAIN.
I	[i,I]	T		U	General index. Multiple use.
IB		C,S_0	INGEOM	R	Signal from INGEOM: 0 or 1. IB=1 denotes a beam.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
IBCE1		C,I,S_0	START	R	Boundary condition end 1. Edge 4 of REPSIL.
IBCE2		C,I,S_0	START	U	Boundary condition end 2. Edge 2 of REPSIL.
IBCE4		(I)	(NONE)	U	REPSIL name. Read into IBCE1. (See input 4)
IENDFR		S_0	ENDFRC	APLFRC	Signal for initial entry into ENDFRC.
IERODE		S_0	ERODE	EROD	Signal for initial entry into ERODE.
IFLAG		T	PDATA	Ū	Control number in subroutine PDATA.
IGAUSS		C,S_0	INGEOM	R	Signal from INGEOM. IGAUSS=1 denotes Gaussian integration through thickness. (If IB=1, IGAUSS=0 in START.)
II		S_0	PDATA	U	Dimension of DAT(I) array. II = 2*NSTRN+8.
IP		C	DGEOM	R	IP=1 denotes mesh point, IP=2 denotes midmesh.
IPRESS		S_0	PRESS	R	Signal for initial entry into PRESS.
IS		C,S_0	START	R	(IS=1 or 2) 1-radial symmetry, 2-slab symmetry.
ISR		C,I,S_0	START	U	Strain rate sensitivity control. (Input 6)
J	[j]	T		U	General index. Occasionally mechanical sublayer.
JCHK(I)		C,I,S_0	START	U	Print controls. (Input 8)
JCYNLP(I)		C,I,S_0	START	U	Time cycles to print LMAT and LMATM. (Input 10)
J1,J2		T	STRAIN	U	Temporary indices equal to NI1(I) and NI2(I).
J3		T	STRAIN	R	Temporary index equal to NI3(I).
K	[k]	T		U	General index. Frequently integration stations.
KD	[k]	D	DGEOM	U	Dummy argument for station K in STRESS and BMSTRS.
KEY		D	(none)	U	Dummy argument in WRTAPE. WRTAPE is not called.
KJ		T	STRESS	U	Index. J'th mechanical sublayer at K'th station.
KJMAX		C,S_0	START	U	Maximum KJ. KJMAX = NSFL*LAYER.
KN		T	STRESS	U	KN = NSFL*(K-1). So, KJ = KN + J. (BMSTRS)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
L	[L]	T	STRESS	U	Number of subincrements for stress increment.
L		T	BMSTRS	R	L=0, or L=1 + INT(4.0*(|SIG22|/SIGYZ - 1.))
L		T		R	General index.
LABELV(I)		I,S_0	ENDFRC	APLFRC	Label for force table. (Input 18)
LABLTE(I)		I,S_0	ERODE	EROD	Label for XEND vs. TEROD table. (Input 19)
LAYER	[L] [K]	C,I,S_0	START	U	Number of integration stations through thickness. (Input 2)
LC		T	STRESS	U	Counter in STRESS. $1 \leq LC \leq L+1$.
LEPSX(I)		S_0	STRAIN	R	Label for EPSX(I) in output. (Output)
LINK		T,D	STRAIN	U	Control switch in STRAIN and PDATA.
LMAT(N,K)		C	STRESS	U	Matrix of LMNK at mesh points. (BMSTRS) (Output)
LMATM(N,K)		C	STRESS	R	Matrix of LMNK at midmeshes. (BMSTRS) (Output)
LMNK		T	STRESS	U	Maximum L, station K, mesh or midmesh N. (BMSTRS)
LOAD		C,I,S_0	START	U	Mode of loading. (Input 5) (DAMP)
LPRESS		C,I,S_0	START	U	Last time step to call PRESS in RPSLID and modify P(N) in DGEOM. (Input 5) (DAMP)
M		T	MOTION	APLFRC	Temporary index. [Not M of REPSIL in Ref. 2.]
MAXC		C,I,S_0	START	U	Final cycle. (Input 3) (DAMP)
MDAMP		C,I,S_0	START	U	Time cycle at which damping begins. (Input 5)
MRITE		C,S_0	START	U	Time cycle to store restart data. (unused)
N	[n]	T		U	Mesh number in ξ^2 direction. $1 \leq N \leq NN < 105$.
NCONT		C,I,S_0	START	U	Initial time step. (Input 3) (Always 0)
NCYC		T	POSITN	U	NCYC = NCYCLE-1.
NCYCH(I)		C,I,S_0	START	U	Cycles for JCHK and energy summary print. (Input 9)
NCYCLE	[ℓ]	C,S	RPSLID	U	Time cycle. (DAMP)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
NC3DP(I)		C,I,S_0	START	U	Time steps for "3D" plots. (Input 11) (DAMP)
ND		D	DGEOM	U	Dummy for N in STRESS, BMSTRS, GRAD, and RESULT.
NDEFX		C,S	STRAIN	R	N corresponding to DEFX.
NDEFXT		T	STRAIN	R	N corresponding to DEFXT.
NDELP		C,S_0	START	U	Time steps between surface strain prints.
NEPSX(I)		C,S	STRAIN	R	Mesh number of EPSX(I).
NEPSXT(I)		T	STRAIN	R	Mesh number of EPSXT(I).
NEST		C,S_0	START	BSTRS	Number of EEB,SSB pairs in table. (NEST<100)
NETAG(I)		C,I,S_0	START	U	Surface for I'th surface strain. (Input 13)
NF		T,I	INVEL	U	Maximum mesh point for uniform velocity. (Input 15)
NFMUS		I,S_0	ENDFRC	APLFRC	Number of EFMUT entries in table. (Input 18)
NFTHS		I,S_0	ENDFRC	APLFRC	Number of EFTHT entries in table. (Input 18)
NTZPTS		I,S_0	ENDFRC	APLFRC	Number of EFZT entries in table. (Input 18)
NI		T,I	INVEL	U	Minimum mesh point for uniform velocity. (Input 17)
NI1(I),NI2(I)		C,S_0	START	U	Mesh points that bracket surface strain points.
NI3(I)		C,S_0	START	R	Midmesh just less than surface strain points.
NLP		C,S	RPSLID	U	Counter for array JCYNLP(I). (DGEOM)
NLPRIN		T,I	START	U	Number of JCYNLP(I) entries. (Input 10)
NMESH		C,I,S_0	START	U	Number of meshes. (Input 2)
NN		C,S_0	START	U	NN = N2B+1. (NN=N2B if IBCE2=4) (NN<105)
NNN		C,S	RPSLID	U	Counter indexing NCYCH(I). (MOTION)
NNPE		C,I,S_0	START	PLOTP	Number of "PLOTP" plots to make. (Input 11a)
NN3D		C,S	PDATA	U	Counter indexing array NC3DP(I).
NOUSE		T	START	R	Position for unwanted input.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
NPE(I)		C,I,S_0	START	PLOTP	N of P(N) at which "PLOTP" plots are desired. IF N>NN, P(N) may be any function of time inserted by programming. (Input 11a)
NPLOT		C,S_0	RPSLID	U	File number for plotting data. (NPLOT = 3)
NPRINT		C,S I	STRAIN START	U U	Next time step for surface strain print. Print interval. (Input 8)
NPTS		T	PDATA	R	Number of points in "3D" plots. NPTS = N2B-N1B+1.
NQ1,NQ2		C,S_0	START	U	Mesh points bracketing ETAD2 location.
NRITE		C,I,S_0	START	U	Restart print interval. (Input 3) (unused)
NSFL	[J]	C,I,S_0	START	U	Number of mechanical stress sublayers. (Input 6)
NSTRN		C,I,S_0	START	U	Number of surface strain locations. (Input 12)
NTEPTS		I,S_0	ERODE	EROD	Number of TEROD,XEND pairs in table. (Input 19)
NUMCY		T,I	START	U	Number of NCYCH(I) entries. (Input 9)
NV		T,I	INVEL	U	Number of points with individual velocities. (Input 15)
N1A,N2A		C,S_0	START (ERODE	R EROD	Limits on N for loop in DGEOM. (Same as N1B,N2B) N1A increases if end point erodes (DE>DECOM).)
N1B,N2B		C,S_0	START (ERODE	R EROD	N at end 1, end 2. N1B = 2. N2B = NMESH+2. N1B increases if end point erodes (DE>DECOM).)
N1V,N2V		C,S_0	START (ERODE	R EROD	Limits on N for points that move. N1V increases if end point erodes (DE>DECOM).)
N3D		T,I	START	U	Number of time steps for "3D" plots. (Input 11)
P(N)	AP,[P,P*]	C,S	PRESS	U	Pressure or force/unit initial area at mesh point N. (DGEOM) (If IB=1, P(N) is force/length.)
PDAT(I)		C,S	PDATA	PLOTP	Array for P(N), N=NPE(I), for plots.
PHIT	ϕ,[ϕ]	T	STRESS	U	Yield function.
PI	π	S_0	STRAIN	U	3.141592653589793. (Too long for CDC)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
PLAST	$[W_p]$	C,S	MOTION	U	Plastic work. PLAST = TNRG-CINET-STREN-TDAMP.
PN(I)	$[n]$	C,S_0	START	U	PN(I) = 2.0 + ETAG2(I)/DETA2.
PN1,PN2		T	START	U	PN1 + NI1(I), PN2 = NI2(I).
PPR(I)		I	PRESS	LINPRS	Pressure at time TPR(I). (Input 16)
PRAT	$E/(1-\nu^2)$	C,S_0	START	U	Material constant used in STRESS.
PSR(J)		C,I,S_0	START	U	Constant for strain rate. (Input 7)
P0		I,S_0	PRESS	R	Pressure in some PRESS subroutines. (Input 16)
QN	$[n]$	C,S_0	START	U	QN = 2 + ETAD2/DETA2. NQ1 ≤ QN < NQ2 = NQ1+1.
QN1,QN2		C,S_0	START	U	Interpolants for displacement at ETAD2.
QS(N)	$\{Q\}$	C,S	RESULT	SHR3/1	Axial force at midmesh N.
Q11,Q22		T	RESULT	U	Proportional to membrane components of stress resultant. (Q11 = SUMA11 - BT*SUMB11)
R(N),Z(N)	$[Y^3,Y^2]$	C,S	INGEOM	R	Coordinate of mesh point N. $\{\underset{\sim}{r} = R\,\underset{\sim}{i}_3 + Z\,\underset{\sim}{i}_2.\}$ [Replaces $\underset{\sim}{r} = Y^j\,\underset{\sim}{i}_j$ of REPSIL] (POSITN)
RADIUS		C,S_0	INGEOM	R	Signals Type of symmetry. If RADIUS>0, radial symmetry. (If IB>0, RADIUS = 0.0 in START.)
RD22	$\{1/(\Delta\xi)^2\}$	C,S_0	START	U	Program constants for differences.
RD22M	$\{½/(\Delta\xi)^2\}$	C,S_0	START	R	
RG	$1/g$	T	STRESS	U	RG = 1.0/DG.
RHO	ρ	T,I	START	U	Mass density. (Input 6)
RMSN	$\{R\}$	T	DGEOM	R	R at current mesh point or midmesh.
RNK,RNR	$\{n_k^-,n_r^-\}$	T	DGEOM	R	Components of normal at time t-Δt. [Replace RN_j]
RSUM		T	START	U	Temporary for computing Δt.
RTD2	$\{½/\Delta\xi\}$	C,S_0	START	U	Constant for differences.
RTD2M	$\{1/\Delta\xi\}$	C,S_0	START	R	Constant for differences.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
RZ(N),ZZ(N)		C,S_0	DGEOM	R	Initial position of mesh point N.
R2,R22	$[Y_2^3, Y_{22}^3]$	C,T	GRAD	R	Difference approximations of $\partial R/\partial\xi$, $\partial^2 R/(\partial\xi)^2$.
SA,SB	$[\alpha,\beta]$	T	STRAIN	U	Temporary constants for surface strains.
SE(J)		C,S_0	START	U	Slope of J'th segment of stress-strain curve.
SEPS(J),SSIG(J)	(ϵ_j,σ_j)	C,I,S_0	START	U	Stress, strain at J'th corner of uniaxial stress-strain curve. (If (e_s,σ_s) is a corresponding engineering stress-strain point, $\epsilon_j = e_s(1+e_s/2)$, $\sigma_j = \sigma_s(1+e_s)$.) (Input 7) $((\epsilon_1,\sigma_1)$ = (SIGZ/E,SIGZ))
SIGYSQ	$(\sigma_0)^2$	T	STRESS	U	Square of dynamic yield stress.
SIGYZ	σ_0	T	BMSTRS	R	Dynamic yield stress.
SIGZ	σ_0	C,I,S_0	START	U	Static uniaxial yield stress. (Input 6)
SIGZSQ(J)	$[(\sigma_{0j})^2]$	C,S_0	START	U	Square of static yield stress, sublayer j.
SIGZZ(J)	$[\sigma_{0j}]$	C,S_0	START	R	Static yield stress, sublayer j. (σ_{0j} = E*SEPS(J))
SIG1(N,KJ), SIG2(N,KJ)	σ^{11},σ^{22}	C,S	STRESS	U	Contravariant components of stress, mesh point N, station K, sublayer J. (BMSTRS,RPSLID)
SIG1M(N,KJ), SIG2M(N,KJ)	σ^{11},σ^{22}	C,S	STRESS	R	Contravariant components of stress, midmesh N, station K, sublayer J. (BMSTRS,RPSLID)
SIG11,SIG22	σ_1^1,σ_2^2	T	STRESS	U	Mixed components of stress. (BMSTRS)
SIG11D,SIG22D	$\sigma_1^{C_1},\sigma_2^{C_2}$	T	STRESS	U	Mixed components of plastic flow corrector stress.
SIG11I,SIG22I	$\sigma^{-1}_1,\sigma^{-2}_2$	T	STRESS	U	Mixed components of stress at time t-Δt. (BMSTRS)
SIG11L,SIG22L	$\sigma_1^{T_1},\sigma_2^{T_2}$	T	STRESS	U	Mixed components of trial stress.
SLABL, SLABW		T,I	INGEOM	R	Length, width of slab symmetric plate. (Input 14)
SNK(N),SNR(N)	(n_k,n_r)	C,S	DGEOM	R	Components of unit normal, mesh point N. (INNORM) [Replace SNi(M,N)]
SNKM(N),SNRM(N)	(n_k,n_r)	C,S	DGEOM	R	Components of unit normal, midmesh N.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
SNKN, SNRN	(n_k, n_r)	T T	DGEOM MOTION	R EAPFRC	Components of unit normal at time t. Components of unit normal at time t+Δt.
SRA	$a^{\frac{1}{2}}$	C,T	DGEOM	U	SRA = SQRT(DA).
SRG	$g^{\frac{1}{2}}$	T	STRESS	U	SRG = SQRT(DG).
SSB(I)	$[\sigma_i]$	C,I,S_0	START	BSTRS	See EEB(I).
SSIG(J)	$[\sigma_j]$	C,I,S_0	START	U	See SEPS(J).
SS1MN(K), SS2MN(K)	σ^{11}, σ^{22}	C	STRESS	U	Contravariant components of stress, station K. (BMSTRS)
SS11, SS22	σ_1^1, σ_2^2	T	STRESS	U	Mixed components of stress for layer.
STREN	[V]	C,S	STRESS	U	Elastic strain energy. (BMSTRS, DGEOM)
SUM		T	START	U	Temporary for computing Δt.
SUMAR		T	START	R	Σ W(K), cross-sectional area for beams.
SUMA11, SUMA22	$\Sigma_0^{11}, \Sigma_0^{22}$	T	RESULT	U	Approximation of $\int_{ZL}^{ZU} \sigma^{\alpha\beta} d\zeta$.
SUMB11, SUMB22	$\Sigma_1^{11}, \Sigma_1^{22}$	T	RESULT	U	Approximation of $\int_{ZL}^{ZU} \zeta\sigma^{\alpha\beta} d\zeta$.
SUMC11, SUMC22	$\Sigma_2^{11}, \Sigma_2^{22}$	T	RESULT	U	Approximation of $\int_{ZL}^{ZU} (\zeta)^2\sigma^{\alpha\beta} d\zeta$.
SUM2AR		T	START	R	$\Sigma_K (ZETA(K))^2 * W(K)$. {Moment of inertia for a beam.}
TA		C,S_0	START	U,R	If IB=1, TA = 1. If IGAUSS = 1, TA = THICKN/2. Otherwise, TA = Δζ = THICKN/ZAYER.
TAMBDA	Δλ	T	STRESS	U	Factor for amount of plastic flow. (Ref. 1, App. C)
TB		T	RESULT	U	Program constant. TB = SRA*TA.
TDAMP	$[W_D]$	C,S	DAMP	U	Damping work. (RPSLID)
TDEFX		C,S	STRAIN	R	Time of DEFX, maximum displacement.
TEMP(N)		C,S_0	DGEOM	U	Program constant for mesh point N. Used in MOTION and KINET. TEMP(N) = DELGAM/$A^{\frac{1}{2}}$.

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
TEMPU		T	BOUNDR	R	Temporary. (BOUNDU)
TEPSX(I)		C,S	STRAIN	R	Time for EPSX(I), the I'th extreme strain.
TEROD(I),XEND(I)		I,S_0	ERODE	EROD	Table of erosion vs. time at end 1. (Input 19)
THICKN	h	T,I	START	U	Thickness of shell. (Input 6)
TIME	t	C,S	RPSLID	U	Time. (START--temporary for output) (DAMP)
TITLE(I)		I	START	U	Array for image of identification. (Input 1)
TMFZ(I)		I,S_0	ENDFRC	APLFRC	Time with EFZT(I). (Input 18)
TMMU(I)		I,S_0	ENDFRC	APLFRC	Time with EFMU(I). (Input 18)
TMTH(I)		I,S_0	ENDFRC	APLFRC	Time with EFTH(I). (Input 18)
TNRG	$\|W\|$	C,S	MOTION	U	Total energy. Initial kinetic energy plus work by external forces.
TPR(I),PPR(I)		I,S_0	PRESS	LINPRS	Time, pressure table for PRESS. (Input 16)
V	$\|v\|$	T,I	INVEL	U	Velocity. (Input 17)
VR	$\|v\|$	T,I	INVEL	U	Velocity. (Input 17)
VR,VZ		T	MOTION	R	Forces, equations of motion. [Replace VM_i+VN_i+VF_i]
VS(N)	$\{V\}$	C,S	RESULT	SHR3/1	Shearing force at midmesh N.
W(K)	$\{W_k\}$	C,S_0	INGEOM START	R R	If IB = 1, W(K) is weight with K'th layer in beam. If IGAUSS=1, W(K) is weight with K'th Gauss point.
WIDTH(K)		T,I	INGEOM	R	Width of K'th layer of beam. (Input 14)
WT(J)		C	START	U	Weight associated with J'th mechanical sublayer.
WZETA(K)	$\{W_{kL}\}$	C,S_0	START	R	Constants for Gaussian integration.
WZETSQ(K)	$\{(W_{kL})^2\}$	C,S_0	START	R	Constants for Gaussian integration.
XEND(I)		I,S_0	ERODE	EROD	See TEROD(I).
YLDFAC		C,I,S_0	START	U	Parameter for plastic flow computation. (Input 2)
Z(N)		C,S	INGEOM	R	See R(N). (POSITN)

NAME	SYMBOL	STATUS	STORED	ORIGIN	COMMENTS
ZAYER		T	START	U	Floating point representation of LAYER.
ZB		T	STRAIN	R	ζ on upper, or lower, surface, (ZU or ZL)
ZETA(K)	$\zeta,(r_k)$	C,S_0 I	START	R R U	If IGAUSS=1, Guassian points through thickness. If IB=1, integration points in beams. (Input 14) Otherwise, midpoints of equal layers.
ZETAK	$[2\zeta]$	T	STRESS	U	2.0*ZETA(K).
ZETASQ(K)	$(\zeta)^2$	C,S_0	START	U	Square of ZETA(K).
ZL	(ζ_ℓ)	C,S_0	START	R	ζ on lower surface. [Replaces -H.]
ZU	(ζ_u)	C,S_0	START	R	ζ on upper surface. [Replaces H.] (Input 14)
ZZ(N)		C,S_0	DGEOM	R	See RZ(N).
Z2,Z22	$[\gamma_2^2,\gamma_{22}^2]$	T	GRAD	R	Approximations for $\partial Z/\partial\xi$, $\partial^2 Z/(\partial\xi)^2$.

APPENDIX C

LISTING OF REPSL1D

The version of RPSL1D discussed in this report is catalogued in file RPSL1D, cycle 2, in UPDATE form at the time of this writing. There is one COMDECK called MAIN. This is listed first. The remainder of the listing is the COMPILE file image of RPSL1D formed through UPDATE with the COMDECK images replaced with the COMMENT statement *CALL MAIN HERE.

This listing gives the correct UPDATE card identifiers. It could be used for simple changes through UPDATE. For involved changes, the user would be wise to use the FTN compiler to create symbolic reference maps.

```
*COMDECK MAIN
      COMMON R(103),Z(103),DR(103),DZ(103),FM11R(103),FM22R(103),       MAIN   2
     1 FM22K(103),FN1TH(103),FN2R(103), FN2K(103),SNR(103),SNK(103),    MAIN   3
     2 TEMP(103),P(103),EPSL1(103),EPSL2(103),EPSU1(103),EPSU2(103),    MAIN   4
     3  SIG1(103,36),SIG2(103,36),LMAT(103,6),                         MAIN   5
     4 DEPS1(6),DEPS2(6),ZETA(6),ZETASQ(6),SS1MN(6),SS2MN(6),W(6),      MAIN   6
     5 NCYCH(50),NC3DP(50),JCYNLP(50)                                   MAIN   7
      COMMON NN,          RD22,RTD2,DETA1,DETA2,RADIUS,NN3D,DAMPF,      MAIN   8
     1 DFACT,NDAMP,TDAMP,LOAD,DELGAM                                    MAIN   9
      COMMON E,ENU,G,PHAT,SIGZ,GAMZ,ZU,ZI,LAYER,DELTAT,TIME,LPRESS,     MAIN  10
     1 NNP,NCYCLE,NWRITE,NCONT,NSTRN,CINER,CINES,CINEP,C1,C2,NPLOT,     MAIN  11
     2 DELSQ,TA,MAXC,NRITE,CA,CB,CINET,STREN,PLAST,TWRG,          EPS,EMP,MAIN  12
     3 NPRINT,NDELP,NMESH,IRCE1,IRCE2,ISW,NSFL,KJMAX,YLDFAC,NLP         MAIN  13
      COMMON N1(6),N2(6),DN1(6),DN2(6),PN(6),FTAG2(6),ANGLE(6),         MAIN  14
     1 ANGLP(6),NETAG(6),EPSS1(6),EPSS2(6),JCHK(3)                      MAIN  15
      COMMON GM,NQ1,NQ2,GN1,GN2,D1,D2,FTAD2                             MAIN  16
      COMMON HSA(6),HSA(6),ASB(6),HSB(6),GI11(6),GI22(6),               MAIN  17
     1 A111(6),A221(6),H111(6),H221(6),A112(6),A222(6),H112(6),H222(6)  MAIN  18
      COMMON DSR(6),HSW(6),SSIG(6),SEPS(6),SF(6),SIGZSQ(6),WT(6)        MAIN  19
      COMMON SIGZZ(6),IH                                                MAIN  20
      COMMON R2,Z2,DR2,DZ2,R22,Z22,DR22,DZ22                            MAIN  21
      COMMON A11,A22,SHA,CS211,CS222,H11,H22,BT,FM11,FM22,CINES1,CINES2 MAIN  22
       COMMON IS                                                        MAIN  23
      COMMON N1B,N2B,N1V,N2V                                            MAIN  24
      COMMON N1A,N2A,IP,RTD2M,RD22M                                     MAIN  25
      COMMONSNRM(103),SNKM(103),SIG1M(103,36),SIG2M(103,36),LMATM(103,6)MAIN  26
      COMMON DEFX,NDEFX,TDEFX,EPSX(8),NEPSX(8),TEPSX(8),                MAIN  27
     1   RZ(103),ZZ(103),GU11(103),GU22(103),GL11(103),GL22(103)        MAIN  28
      COMMON DR3(6),N13(6),DA1(103),DR1(103),DA2(103),DR2(103)          MAIN  29
       COMMON IGAUSS,WZETA(6),WZETSQ(6)                                 MAIN  30
```

```
      PROGRAM RPSL10 (INPUT,OUTPUT,PLOT,TAPE5=INPUT,TAPE6=OUTPUT,          RPSL10    2
     1  TAPE3=PLOT)                                                        RPSL10    3
C                    *CALL MAIN HERE                                      TAPE      1
C                                                                         RPSL10    5
C        CHANGES IN SURFACE STRAIN COMPUTATIONS 5/26/76                   RPSL10    6
C        OPTIONAL GAUSSIAN INTEGRATION 6/9/76.  USED IF IGAUSS=1.         RPSL10    7
C        PERMIT UP TO 6 MECHANICAL SUBLAYERS.  NSFL  .LE. 6               RPSL10    8
C                    (1-D)REPSIL MASTER PROGRAM FOR CDC.  10/13/77        RPSL10    9
C        INDEX FOR BEAM OR SLAB SYMMETRY.  PRESS FOR CONSTANT PRESSURE.   RPSL10   10
C        IRCE1=1,2,OR3.  IRCE2=1,2,3,OR4.  6/28/76  1/7/76                RPSL10   11
C        CHANGES TO PERMIT SYMMETRY AT END 1 INSERTED ON 1/7/76           RPSL10   12
C        PERMIT UP TO 4 MECHANICAL SUBLAYERS.  NSFL  .LE. 4.              RPSL10   13
C        END2 FREE IF IRCE2=4.  8/28/76.  MAY ONLY BE GOOD FOR BEAMS.     RPSL10   14
C        DIFFERENCES FOR FM CENTERED AT MESH POINTS.  FOR FN IN MESHES    RPSL10   15
C             TAPE (NPLOT) PLOTTING DATA                                  RPSL10   16
C                                                                         RPSL10   17
C             INITIATE PROGRAM                                            RPSL10   18
C                                                                         RPSL10   19
      DO 901 L=1,4                                                        RPSL10   20
      EPSX(L)=-100000.                                                    RPSL10   21
  901 EPSX(L+4)=100000.                                                   RPSL10   22
      DEFX=0.0                                                            RPSL10   23
C                                                                         RPSL10   24
      NPLOT=3                                                             RPSL10   25
      REWIND 3                                                            RPSL10   26
      NLP=1                                                               RPSL10   27
      NNN=1                                                               RPSL10   28
      CALL START                                                          RPSL10   29
C                                                                         RPSL10   30
C             DELETE RESTART.  (I.E. WRTAPE)                              RPSL10   31
C     IF(NCONT .LE. 0) GOTO 15                                            RPSL10   32
C     CALL WRTAPE (2)                                                     RPSL10   33
C     NPRINT=(NCYCLE-MOD(NCYCLE,NDELP))+NDELP                             RPSL10   34
C     GOTO 49                                                             RPSL10   35
   15 NCYCLE=0                                                            RPSL10   36
      TIME=0.0                                                            RPSL10   37
      CINES=0.0                                                           RPSL10   38
      TDAMP=0.0                                                           RPSL10   39
      CINEP=0.0                                                           RPSL10   40
      ENS=0.0                                                             RPSL10   41
      D1=0.0                                                              RPSL10   42
      D2=0.0                                                              RPSL10   43
      [illegible]                                                         RPSL10   44
C                                                                         RPSL10   45
C     SET INITIAL DISPLACEMENT,PRESSURE,STRAIN AND STRESS = 0             RPSL10   46
C                                                                         RPSL10   47
   17 DO 28 N=1,NN                                                        RPSL10   48
      DR(N)=0.0                                                           RPSL10   49
      DZ(N)=0.0                                                           RPSL10   50
      P(N)=0.0                                                            RPSL10   51
      EPSL1(N)=0.0                                                        RPSL10   52
      EPSL2(N)=0.0                                                        RPSL10   53
      EPSU1(N)=0.0                                                        RPSL10   54
      EPSU2(N)=0.0                                                        RPSL10   55
      DO 28 K=1,KJMAX                                                     RPSL10   56
      SIG1(N,K)=0.0                                                       RPSL10   57
      SIG2(N,K)=0.0                                                       RPSL10   58
      SIG1M(N,K)=0.0                                                      RPSL10   59
      SIG2M(N,K)=0.0                                                      RPSL10   60
   28 CONTINUE                                                            RPSL10   61
C                                                                         RPSL10   62
      [illegible] CALL PRESS                                              RPSL10   63
```

```
      DELGAM=DELSQ/GAMZ
      CALL DGFOR
      CALL STRAIN
      DO 33 N=N1R,N2R
      A22 = 0.5*(DA2(N)+DA2(N-1))
      IF(N .EQ. N1R)A22=DA2(N)
      IF(N .EQ. N2R)A22=DA2(N-1)
      GU11(N) = 1.0/(DA1(N)-2.0*ZU*DB1(N))
      GU22(N) = 1.0/(     A22-2.0*ZU*DB2(N))
      GL11(N) = 1.0/(DA1(N)-2.0*ZL*DB1(N))
      GL22(N) = 1.0/(     A22-2.0*ZL*DB2(N))
      DA1(N) = 0.0
      DA2(N-1) = 0.0
      DB1(N) = 0.0
      DB2(N) = 0.0
   33 CONTINUE
      DA2(N2R) = 0.0
      EPSU2(N2R) = 0.0
      EPSL2(N2R) = 0.0
      IF(LOAD .GT. 0)GOTO 43
C
      CALL INVEL
C
      DO 35 N=N1V,N2V
      DR(N)=DELTAT*DR(N)
      DZ(N)=DELTAT*DZ(N)
   35 CONTINUE
      CALL BOUNDU
      CALL KINET
      CINES=2.0*CINET
      TNRG=CINES
C
      IF(LOAD) 42,45,43
C
   42 CALL PWORK
C
   43 DO 44 N=N1V,N2V
       DR(N)=DR(N)-P(N)*SNR(N)*TEMP(N)
       DZ(N)=DZ(N)-P(N)*SNZ(N)*TEMP(N)
   44 CONTINUE
      CALL BOUNDU
      CALL PWORK
      ENR=ENS
      CALL KINET
      TNRG=CINET
   45 IF(NCYCH(1) .EQ. 0)NNN=2
C         WRITE INITIAL CARTESIAN COORDINATES, PRESSURE
      WRITE(6,300)
      WRITE(6,400)(N,R(N),Z(N),P(N),N=1,NN)
C
   49 CALL PDATA (1)
C
C      END INITIALIZATION
C
   50 NCYCLE=NCYCLE+1
      TIME=TIME+DELTAT
C      CHECK FOR FINAL STEP     .
      IF(NCYCLE .GT. MAXC)GOTO 70
C         CHECK IF CALL PRESS IS NEEDED
      IF(LPRESS .GE. NCYCLE)CALL PRESS
      CALL POSITN
      CALL [illegible]
      CALL STRAIN
```

```
      CALL MOTION                                                       RFSL1D     127
      CALL PDATA (2)                                                    RFSL1D     128
      CALL DAMP                                                         RFSL1D     129
C           CHECK FOR RESTART DUMP                                      RFSL1D     130
C     IF(NCYCLE .NE. NRITE)GOTO 50                                      RFSL1D     131
C     CALL WRTAPE (1)                                                   RFSL1D     132
C     NRITE=NRITE+NRITE                                                 RFSL1D     133
C     CALL PDATA (3)                                                    RFSL1D     134
      GOTO 50                                                           RFSL1D     135
   70 IF(NCYCLE .LT. 2)STOP' MAIN.  NCYCLE .LT.                         RFSL1D     136
      CALL PDATA (3)                                                    RFSL1D     137
      CALL PDATA (4)                                                    RFSL1D     138
      STOP' PROGRAM COMPLETE'                                           RFSL1D     139
  300 FORMAT(1H1,20X,19HINITIAL COORDINATES,23X,8HPRESSURE//            RFSL1D     140
     1 3X,1HN,11X,4HR(N),21X,4HZ(N),21X,4HP(N))                         RFSL1D     141
  400 FORMAT((I4,3(2X,E23.16)))                                         RFSL1D     142
      END                                                               RFSL1D     143
      SUBROUTINE START                                                  START        2
C                     *CALL MAIN HERE                                   TAF          1
      DIMENSION TITLE(8)                                                START        4
      READ(5,100) TITLE                                                 START        5
      WRITE(6,140) TITLE                                                START        6
      READ(5,105)NOUSE,NMESH,LAYER,YLDFAC                               START        7
      READ(5,105) MAXC,NCONT,NRITE,DELTAT                               START        8
      READ(5,110) NOUSE,IBCE2,NOUSE,IBCE1                               START        9
      READ(5,115) LOAD,LPRESS,NDAMP,DAMPF,DFACT                         START       10
      READ(5,120) E,ENU,SIGZ,RHO,THICKN,NSFL,ISR                        START       11
      WRITE(6,160) E,ENU,SIGZ,RHO,THICKN                                START       12
      WRITE(6,170) NCONT,MAXC,NPRINT,NRITE                              START       13
      WRITE(6,175) LAYER,NSTRN,LOAD,LPRESS                              START       14
      WRITE(6,180) IBCE1,IBCE2                                          START       15
      IF(NSFL .EQ. 1 .AND. ISR .EQ. 0)GOTO 700                          START       16
      IF(NSFL .EQ. 0 )GOTO 700                                          START       17
      READ (5,125) (SSIG(J),SEPS(J),DSR(J),PSR(J),J=1,NSFL)             START       18
  700 IF(NSFL .LT. 1) ISR=-1                                            START       19
      IF(NSFL .LT. 1) NSFL=1                                            START       20
      READ(5,110) NPRINT,(JCHK(J),J=1,3)                                START       21
      READ(5,110) NUMCY,(NCYCH(J),J=1,NUMCY)                            START       22
      READ(5,110) NLPRIN,(JCYNLP(J),J=1,NLPRIN)                         START       23
      READ(5,110) N3D,(NC3DP(J),J=1,N3D)                                START       24
      SSIG(1)=SIGZ                                                      START       25
      SEPS(1)=SIGZ/E                                                    START       26
      KJMAX=LAYER*NSFL                                                  START       27
      SE(1)=E                                                           START       28
      SIGZSQ(1)=SSIG(1)**2                                              START       29
      SIGZZ(1) = SSIG(1)                                                START       30
      DO 795 J=1,NSFL                                                   START       31
      IF(ISR .LT. 1)GOTO 794                                            START       32
      IF(DSR(J) .GT. 0.0 .AND. PSR(J) .GT. 0.0)GOTO 793                 START       33
  791 WRITE(6,792)                                                      START       34
  792 FORMAT(//46H ERROR IN STRAIN HARDENING OR STRAIN RATE DATA  )     START       35
      STOP' START.  ERROR IN STRAIN DATA'                               START       36
  793 PSR(J) = 1.0/PSR(J)                                               START       37
  794 IF(J .EQ. 1) GOTO 795                                             START       38
      IF(SEPS(J).LE.SEPS(J-1)) GOTO 791                                 START       39
      SE(J)=(SSIG(J)-SSIG(J-1))/(SEPS(J)-SEPS(J-1))                     START       40
      WT(J-1)=(SE(J-1)-SE(J))/E                                         START       41
      SIGZSQ(J)=(E*SEPS(J))**2                                          START       42
      SIGZZ(J)=(E*SEPS(J))                                              START       43
  795 CONTINUE                                                          START       44
      WT(NSFL)=SE(NSFL)/E                                               START       45
C     SET UP BOUNDARY CONTROLS                                          START       46
C         IBCE1=1,2,OR3.  IBCE2=1,2,3,OR4.   6/22/76   1/7/76           START       47
```

```
      N1H=2                                                                  START     48
      N2P=NMESH+N1H                                                          START     49
      N1V=N1H+1                                                              START     50
      N2V=N2H+1                                                              START     51
      N1A=N1V-1                                                              START     52
      N2A=N2V+1                                                              START     53
      IF(IBCE1 .EQ. 2)N1V=N1H                                                START     54
      IF(IBCE2.EQ.2.OR.IBCE2.EQ.4)N2V=N2H                                    START     55
      NN=N2A+1                                                               START     56
       IF(IH(F2 .EC. 4) NN=N2H                                               START     57
      READ(5,130) NOUSE,ETAD2,NSTRN                                          START     58
      READ(5,15)    (NOUSE,ETAG2(I),ANGLE(I),ANGLE(I),NETAG(I),I=1,NSTRN)    START     59
      CALL INGEOM                                                            START     60
      CALL INMOPM                                                            START     61
C        SIGNALS TO PROGRAM FROM INGEOM.                                      START     62
C          AXIAL SYMMETRY  (RADIUS > 0.0,  IF=0).                             START     63
C           SLAB SYMMETRY  (RADIUS = 0.0   IF=0).     BEAM (IH>0).           START     64
      IS = 1                                                                 START     65
      IF(IB .GT. 0) RADIUS = 0.0                                             START     66
      IF(RADIUS .LE. 0.0) IS=2                                               START     67
      IF(IB .GT. 0)DETA1=1.0                                                 START     68
      GN=FLOAT(N1H) *ETAD2/DETA2                                             START     69
      NG1=GN                                                                 START     70
      NG2=NG1+1                                                              START     71
      GN1=GN-FLOAT(NG1)                                                      START     72
      GN2=FLOAT(NG2)-GN                                                      START     73
      DO 20 I=1,NSTRN                                                        START     74
      PN(I)=FLOAT(N1H) *ETAG2(I)/DETA2                                       START     75
      NI1(I)=PN(I)                                                           START     76
      NI2(I)=NI1(I)+1                                                        START     77
      PN1=NI1(I)                                                             START     78
      PN2=NI2(I)                                                             START     79
      DN1(I)=PN(I)-PN1                                                       START     80
      DN2(I)=PN2-PN(I)                                                       START     81
      IF(DN1(I) .GT. 0.5)GOTO 16                                             START     82
C        DN1(I) LE 1/2.                                                      START     83
      NI3(I) = NI1(I)-1                                                      START     84
      DN3(I) = DN1(I) + 0.5                                                  START     85
      GOTO 17                                                                START     86
C        DN1(I) GT 1/2                                                       START     87
   16 NI3(I) = NI1(I)                                                        START     88
      DN3(I) = DN1(I) - 0.5                                                  START     89
   17 IF(NI3(I) .GT. 1)GOTO 18                                               START     90
C        NI3(I) LT 2.  SHIFT TO 2.                                           START     91
      NI3(I) = 2                                                             START     92
      DN3(I) = 0.0                                                           START     93
      GOTO 20                                                                START     94
   18 IF(NI3(I) .LT. N2H-1) GOTO 20                                          START     95
C        NI3(I) GT N2H-2.  SHIFT TO N2H-2.                                   START     96
      NI3(I) = N2H-2                                                         START     97
      DN3(I) = 1.0                                                           START     98
   20 CONTINUE                                                               START     99
      ZAYER=LAYER                                                            START    100
C              PROGRAM CONSTANTS                                             START    101
      IF(IH .LE. 0) Z0=0.5*THICKN                                            START    102
      ZL=Z0-THICKN                                                           START    103
      GAMZ=THICKN*PHO                                                        START    104
      NDELP=NPRINT                                                           START    105
      MWITE=NCONT*NWRITE                                                     START    106
      SUMAR=0.0                                                              START    107
      SUM2AR=0.0                                                             START    108
C        IGAUSS=1 IS SIGNAL FOR GAUSSIAN INTEGRATION THRU THICKNESS.         START    109
      IF(IH .EQ. 1)IGAUSS=0                                                  START    110
```

```
      IF(IGAUSS .NE. 1)GOTO 50                                          START   111
C        SET UP FOR GAUSSIAN INTEGRATION WITH LAYER (.LE.6) POINTS       START   112
      GOTO(41,42,43,44,45,46),LAYER                                     START   113
   41 ZETA(1) = 0.0                                                     START   114
      W(1) = 2.0                                                        START   115
      GOTO 50                                                           START   116
   42 ZETA(2) = 0.577350269189626*ZU                                    START   117
      ZETA(1) = -ZETA(2)                                                START   118
      W(1) = 1.0                                                        START   119
      W(2) = 1.0                                                        START   120
      GOTO 50                                                           START   121
   43 ZETA(3) = 0.774596669241483*ZU                                    START   122
      ZETA(2) = 0.0                                                     START   123
      ZETA(1) = -ZETA(3)                                                START   124
      W(1) = 0.555555555555556                                          START   125
      W(2) = 0.888888888888889                                          START   126
      W(3) = W(1)                                                       START   127
      GOTO 50                                                           START   128
   44 ZETA(4) = 0.861136311594053*ZU                                    START   129
      ZETA(3) = 0.339981043584856*ZU                                    START   130
      ZETA(2) = -ZETA(3)                                                START   131
      ZETA(1)= -ZETA(4)                                                 START   132
      W(1) = 0.347854845137454                                          START   133
      W(2) = 0.652145154862546                                          START   134
      W(3) = W(2)                                                       START   135
      W(4) = W(1)                                                       START   136
      GOTO 50                                                           START   137
   45 ZETA(5) = 0.906179845938664*ZU                                    START   138
      ZETA(4) = 0.538469310105683*ZU                                    START   139
      ZETA(3) = 0.0                                                     START   140
      ZETA(2) = -ZETA(4)                                                START   141
      ZETA(1) = -ZETA(5)                                                START   142
      W(1) = 0.236926885056189                                          START   143
      W(2) = 0.478628670499366                                          START   144
      W(3) = 0.568888888888889                                          START   145
      W(4) = W(2)                                                       START   146
      W(5) = W(1)                                                       START   147
      GOTO 50                                                           START   148
   46 ZETA(6) = 0.932469514203152*ZU                                    START   149
      ZETA(5) = 0.661209386466265*ZU                                    START   150
      ZETA(4) = 0.238619186083197*ZU                                    START   151
      ZETA(3) = -ZETA(4)                                                START   152
      ZETA(2) = -ZETA(5)                                                START   153
      ZETA(1) = -ZETA(6)                                                START   154
      W(1) = 0.171324492379170                                          START   155
      W(2) = 0.360761573048139                                          START   156
      W(3) = 0.467913934572691                                          START   157
      W(4) = W(3)                                                       START   158
      W(5) = W(2)                                                       START   159
      W(6) = W(1)                                                       START   160
   50 CONTINUE                                                          START   161
      DO 3 K=1,LAYER                                                    START   162
      IF(IGAUSS .EQ. 1)GOTO 51                                          START   163
      IF(JP .LE. 0)ZETA(K)=ZU*(1.0-(2.0*FLOAT(K)-1.0)/ZAYER)            START   164
   51 CONTINUE                                                          START   165
      SUMAR=SUMAR+W(K)                                                  START   166
      ZETASQ(K) = ZETA(K)**2                                            START   167
      SUM2AR=SUM2AR+W(K)*ZETASQ(K)                                      START   168
      WZETA(K) = W(K)*ZETA(K)                                           START   169
      WZETSQ(K) = ZETA(K)*ZETA(K)                                       START   170
    3 CONTINUE                                                          START   171
      TA=THICKN/ZAYER                                                   START   172
      IF(JP .NE. 0)TA=1.0                                               START   173
```

```
      IF(IGAUSS .EQ. 1)TA=ZU
      IF(IH .GT. 0)GAMZ=RHO*SUMAH
C          TIME INCREMENT BY VON NEUMAN
      C=SQRT(E/RHO)
      IF(IB .LE. 0) C=C/SQRT(1.0-ENU**2)
      DELM=DETA2/C
      CTWO=(1.0-1.0/ZAYER**2)*THICKN**2/12.0
      IF(IH.GT.0)CTWO=SUM2AH/SUMAH
      DELB= 0.5*DETA2**2/(C*SQRT(CTWO))
      IF(IGAUSS .EQ. 1)DELB = SQRT(3.)*DETA2**2/(C*THICKN)
      DELMIN=AMIN1(DELB,DELM)
      DELMIN= 0.95*AMIN1(DELB,DELM)
      SUM=1.0
   25 RSUM=1.0/SUM
      SUM=SUM*10.0
      IF(RSUM .LT. DELMIN)GOTO 30
      GOTO 25
   30 DELMIN=AINT(DELMIN*SUM)/SUM
      DELIN=DELTAT
      IF(DELIN .GT. 0.0)DELMIN=AMIN1(DELIN,DELMIN)
      DELTAT=DELMIN
      DELSQ=DELTAT**2
      G=.5*E/(1.0+ENU)
      EBAT=E/(1.0-ENU**2)
      HTO2=1.0/(2.0*DETA2)
      HO22=1.0/DETA2**2
      HTO2M=1.0/DETA2
      HO22M=0.5/DETA2**2
      CA=DETA1*DETA2
      CB = 0.5*TA*DETA1*DETA2/E
C          DAMPING CONSTANTS
      C2=2.0*DELTAT*DAMPF/GAMZ
      C1=C2/(4.0+C2)
C
      WRITE(6,130)
      WRITE(6,140) TITLE
      WRITE(6,150)    DETA1,    NMESH,DETA2
      WRITE(6,300) DELB,DELM,DELIN,DELTAT
      WRITE(6,160) E,ENU,SIGZ,RHO,THICKN
      WRITE(6,170) NCONT,MAXC,NPRINT,NRITE
      WRITE(6,175) LAYER,NSTRN,LOAD,LPRESS
      WRITE(6,180) IHCE1,IHCE2
      WRITE(6,185) (JCHK(I),I=1,3)
      WRITE(6,190) (NCYCP(I),I=1,NNCY)
      WRITE(6,500) (JCYNLP(J),J=1,NLPRIN)
      WRITE(6,195) (NC3DP(I),I=1,N3D)
      IF(ISH .EQ. -1) WRITE(6,400)
      IF(NSFL .EQ. 1 .AND. ISH .EQ. 1) WRITE(6,405)
      IF(NSFL .GT. 1 .AND. ISH .EQ. 0) WRITE(6,410)
      IF(NSFL .EQ. 1 .AND. ISH .EQ. 0) WRITE(6,415)
      IF(NSFL .GT. 1 .AND. ISH .EQ. 1) WRITE(6,420)
      IF(NSFL.GT.1)WRITE(6,820)NSFL
      WRITE(6,821)(J,SSIG(J),SEPS(J),DSP(J),PSH(J),J=1,NSFL)
      WRITE(6,110)
      TIME=DELTAT*FLOAT(NDAMP)
      WRITE(6,200) NDAMP,TIME,DAMPF,DFACT
      IF(IHCE1 .LT. 4 .AND. IHCE2 .LT. 5)RETURN
      WRITE(6,96474)
      STOP' START, BAD END DEFINITION'
96474 FORMAT(20X, 'END CONDITIONS NOT ALLOWED')
   15 FORMAT(4F10.4,I5)
   [illegible]
  105 FORMAT(3I5,E12.6)
```

```
  110 FORMAT(16I5)                                                      START    237
  115 FORMAT(3I5,2F12.6)                                                START    238
  120 FORMAT(5E12.6,2I5)                                                START    239
  125 FORMAT(4E15.7)                                                    START    240
  130 FORMAT(2E10.4,I5)                                                 START    241
  140 FORMAT(1H1,53X,15HREAL EPSILON CODE//24X,8A10/)                   START    242
  150 FORMAT(43X,34HCOMPUTATIONS FOR ENERGY USE DETA1=,E12.6/           START    243
     1       3HX,I4,25H MESHES IN ETA2 DIRECTION,3X,7H(DETA2=F12.6,1H)/) START    244
  160 FORMAT(32X,17HYOUNG'S MODULUS =E12.6                              START    245
     1      /32X,17HPOISSON'S RATIO =E12.6,10X,17HYIELD STRESS     =E12.6 START    246
     2      /32X,17HMASS DENSITY    =E12.6,10X,17HTHICKNESS        =E12.6 START    247
     3/)                                                                START    248
  170 FORMAT(55X,18HSTART AT TIME STEP=I5/55X,18HFINAL  TIME  STEP =I5/  START    249
     1       55X,21HSURFACE STRAINS EVERY=I5,10H TIME STEP/             START    250
     2       55X,21HRESTART  WRITE  EVERY=I5,10H TIME STEP/)            START    251
  175 FORMAT(43X,7HLAYER =I5,18X,8HNSTRN  =I5/                          START    252
     1       43X,7HLOAD  =I5,1HX,8HLPRESS =I5/)                         START    253
  180 FORMAT(54X,'BOUNDARY CONDITIONS'/24X,'1/2/3/4/     = CLAMPED/'    START    254
     1 'SYMMETRY/HINGED/FREE/                          '/               START    255
     2 50X,'END1 (IBCE1) =',I4/50X,'END2 (IBCE2) =',I4/)                START    256
  185 FORMAT(50X,25HPRINT OPTION CONTROL CARD/52X,20H0/1 = NO PRINT/PRIN START    257
     1T/50X,I4,24H DISPLACEMENT INCREMENTS/                             START    258
     2  50X,I4,32H CARTESIAN COORDINATES, PRESSURE/                     START    259
     3  50X,I4,33H SURFACE NORMAL VECTOR COMPONENTS/)                   START    260
  190 FORMAT(24X,45HPRINT INFORMATION AT THE FOLLOWING TIME STEPS/(24X,  START    261
     1(16I5)))                                                          START    262
  195 FORMAT(24X,3HB3=0 PLOTS FOR THE FOLLOWING TIME STEPS/(24X,(16I5))) START    263
  200 FORMAT(/46X,29HSTART DAMPING AFTER TIME STEP=I5,5X,6HTIME =E10.4/ START    264
     140X,7HDAMPF =F10.4,16X,7HDFACT =F10.4)                            START    265
  300 FORMAT(47X,25HBEGINNING  TIME INCREMENT= E12.6/47X,25HMEMBRANE TIME START    266
     1INCREMENT= E12.6/47X,25HINPUT    TIME INCREMENT= E12.6//44X,31HTIM START    267
     2E INCREMENT USED BY REPSIL= F12.6/)                               START    268
  400 FORMAT(/37X,32HCONSTITUTIVE RELATION     ELASTIC)                 START    269
  405 FORMAT(/37X,78HCONSTITUTIVE RELATION     ELASTOPLASTIC-NO WORK HARD START    270
     1ENING-STRAIN RATE DEPENDENT)                                      START    271
  410 FORMAT(/37X,77HCONSTITUTIVE RELATION     ELASTOPLASTIC-WORK HARDENI START    272
     1NG-STRAIN RATE INDEPENDENT)                                       START    273
  415 FORMAT(37X,80HCONSTITUTIVE RELATION     ELASTOPLASTIC-NO WORK HARDE START    274
     1NING-STRAIN RATE INDEPENDENT)                                     START    275
  420 FORMAT(37X,75HCONSTITUTIVE RELATION     ELASTOPLASTIC-WORK HARDENIN START    276
     1G-STRAIN RATE DEPENDENT)                                          START    277
  500 FORMAT(24X,49HPRINT L MATRIX (LMAT) AT THE FOLLOWING TIME STEPS/  START    278
     1(24X,16I5))                                                       START    279
  820 FORMAT(35X,31HSTRESS-STRAIN APPROXIMATION HAS,I3,10H SUBLAYERS)   START    280
  821 FORMAT(/46X,40HSTRESS-STRAIN AND STRAIN RATE PARAMETERS/          START    281
     130X,1HJ,9X,7HSSIG(J),8X,7HSEPS(J),14X,6HDSH(J),8X,8H1/DSP(J)/     START    282
     2(26X,I5,5X,1P2E15.7,5X,2E15.7))                                   START    283
      END                                                               START    284
      SUBROUTINE INVEL                                                  INVEL      2
C                    *CALL MAIN HERE                                    TAP        1
C       EVALUATE THE INITIAL VELOCITY AT TIME=0 FOR ALL MESH POINTS     INVEL      4
C          INPUT INDICES RELATIVE TO END1 POINT = 1                     INVEL      5
      READ(5,100)NI,NF,VR,NV                                            INVEL      6
      NI=NI+NI-1                                                        INVEL      7
      NF=NF+NI-1                                                        INVEL      8
      WRITE(6,200) NI,NF,VR                                             INVEL      9
C                                                                       INVEL     10
      DO 30 N=NI,NF                                                     INVEL     11
      DR(N)=-VR*SNR(N)                                                  INVEL     12
      DZ(N)=-VR*SNZ(N)                                                  INVEL     13
   30 CONTINUE                                                          INVEL     14
      IF(NV .LE. 0)GO TO 50                                             INVEL     15
      WRITE(6,300)                                                      INVEL     16
```

```
      DO 45 K=1,NV                                                      INVEL     17
      READ(5,500)N,V                                                    INVEL     18
       N= N+NP1-1                                                       INVEL     19
      WRITE(6,400)N,V                                                   INVEL     20
      DR(N)=-V*SNR(N)                                                   INVEL     21
      DZ(N)=-V*SNZ(N)                                                   INVEL     22
   45 CONTINUE                                                          INVEL     23
   50 RETURN                                                            INVEL     24
C                                                                       INVEL     25
  100 FORMAT(10X,2I5,F12.6,I5)                                          INVEL     26
  200 FORMAT(1H1,40X,2HN=,I3,' TO',I3, 15H VELOCITY (VH)=,E12.6/)       INVEL     27
  300 FORMAT(52X,27HOTHER VELOCITY DISTRIBUTION/63X,1HN,8X,1HV/)        INVEL     28
  400 FORMAT(59X,I5,2X,E23.16)                                          INVEL     29
  500 FORMAT(5X,I5,F12.6)                                               INVEL     30
      END                                                               INVEL     31
      SUBROUTINE POSITN                                                 POSITN     2
C                    *CALL MAIN HERE                                    TAE        1
      DO 50 N=1,NN                                                      POSITN     4
      R(N)=R(N)+DR(N)                                                   POSITN     5
      Z(N)=Z(N)+DZ(N)                                                   POSITN     6
   50 CONTINUE                                                          POSITN     7
      IF(NCYCLE .NE. NCYCH(NPN))GOTO 75                                 POSITN     8
      IF(JCHK(1).LE.0.AND.JCHK(2).LE.0)GOTO 75                          POSITN     9
C       WRITE DISPLACEMENT INCREMENTS, COORDINATES, AND PRESSURE        POSITN    10
                                                                        POSITN    11
      NCYC=NCYCLE-1                                                     POSITN    12
      WRITE(6,101)NCYCLE,TIME                                           POSITN    13
      WRITE(6,102)NCYC,NCYCLE                                           POSITN    14
      WRITE(6,103)                                                      POSITN    15
      WRITE(6,100)(N,DR(N),DZ(N),R(N),Z(N),P(N),N=1,NN)                 POSITN    16
                                                                        POSITN    17
   75 RETURN                                                            POSITN    18
  100 FORMAT((I5,5(2X,F23.16)))                                         POSITN    19
  101 FORMAT(10HITIME STEP,I5,4X,4HTIME,F16.8)                          POSITN    20
  102 FORMAT(/6X,36HDISPLACEMENT INCREMENTS BETWEEN T.S.,I4,4H AND,I4,  POSITN    21
     1 21X,11HCOORDINATES,27X,8HPRESSURE/)                              POSITN    22
  103 FORMAT(4X,1HN,12X,5HDR(N),20X,5HDZ(N),21X,4HR(N),21X,4HZ(N),21X,4H POSITN    23
     1P(N))
      END                                                               DGEOM      2
      SUBROUTINE DGEOM                                                  TAE        1
C                    *CALL MAIN HERE                                    DGEOM      4
      STREN=0.0                                                         DGEOM      5
      A11=1.0                                                           DGEOM      6
      R11=0.0                                                           DGEOM      7
      CS21=0.0                                                          DGEOM      8
      [illegible]=0.0                                                   DGEOM      9
      DR11=0.0                                                          DGEOM     10
      DO 91 IP=1,2                                                      DGEOM     11
C        IP=1 DENOTES MESH POINTS,  IP=2 DENOTES MESHES                 DGEOM     12
      DO 90 N=N1A,N2A                                                   DGEOM     13
      IF(IP .EQ. 2)GOTO 12                                              DGEOM     14
      RMSN=R(N)                                                         DGEOM     15
      DRMSN=DR(N)                                                       DGEOM     16
      RNK=SNR(N)                                                        DGEOM     17
      RNK=SNK(N)                                                        DGEOM     18
      GOTO 13                                                           DGEOM     19
   12 RNK=SNRM(N)                                                       DGEOM     20
      RNK=SNKM(N)                                                       DGEOM     21
C        AVERAGE RADIUS AND CHANGE OF RADIUS, IN MESH                   DGEOM     22
      RMSN=0.5*(R(N)+R(N+1))                                            DGEOM     23
      DRMSN=0.5*(DR(N)+DR(N+1))                                         DGEOM     24
   13 CONTINUE                                                          DGEOM     25
      CALL GRAD(N)                                                      DGEOM     26
      IF(IS .EQ. 1)A11=RMSN**2                                          DGEOM     27
```

```
      L=SQRT(A22)                                                       DGEOM   28
      DA=A11*A22                                                        DGEOM   29
      SRA=SQRT(DA)                                                      DGEOM   30
      SNRN=Z22/L                                                        DGEOM   31
      SNKN=-R22/L                                                       DGEOM   32
      IF(IP .EQ. 2)GOTO 22                                              DGEOM   33
C        IP=1 DENOTES MESH POINTS.  IP=2 DENOTES RESHES                 DGEOM   34
      IF(LPRESS .GE. NCYCLE)R(N)=SRA*R(N)                               DGEOM   35
      IF(N .EQ. N1P .AND. IHOLE1 .EQ. 1)GOTO 20                         DGEOM   36
      IF(N .EQ. N2P .AND. IHOLE2 .EQ. 1)GOTO 20                         DGEOM   37
      SNR(N)=SNRN                                                       DGEOM   38
      SNK(N)=SNKN                                                       DGEOM   39
      GOTO 23                                                           DGEOM   40
   20 SNRN=SNR(N)                                                       DGEOM   41
      SNKN=SNK(N)                                                       DGEOM   42
      GOTO 23                                                           DGEOM   43
   22 SNRM(N)=SNRN                                                      DGEOM   44
      SNKM(N)=SNKN                                                      DGEOM   45
   23 CONTINUE                                                          DGEOM   46
      IF(IS .EQ. 1)R11=-SNRN*RMSN                                       DGEOM   47
      R22=SNRN*R22+SNKN*Z22                                             DGEOM   48
      IF(NCYCLE.GT.0)GOTO 88                                            DGEOM   49
      DA2(N) = A22                                                      DGEOM   50
      IF(IP .EQ. 2)GOTO 90                                              DGEOM   51
      TESR(N)=DELGAM/SRA                                                DGEOM   52
      DA1(N) = A11                                                      DGEOM   53
      DR1(N) = R11                                                      DGEOM   54
      DR2(N) = R22                                                      DGEOM   55
      RZ(N)=R(N)                                                        DGEOM   56
      ZZ(N)=Z(N)                                                        DGEOM   57
      GOTO 90                                                           DGEOM   58
   88 AR11=A22/DA                                                       DGEOM   59
      AR22=A11/DA                                                       DGEOM   60
      RR11=AR11*R11                                                     DGEOM   61
      RR22=AR22*R22                                                     DGEOM   62
      RT=RR11+RR22                                                      DGEOM   63
      IF(IS .EQ. 1)CS211=-RMSN*R2*AR22                                  DGEOM   64
      CS222=(R2*R22+Z2*Z22)*AR22                                        DGEOM   65
      DNL2=-(RNR*DR2+RNK*DZ2)                                           DGEOM   66
      DNR2=DNL2*AR22                                                    DGEOM   67
      DN=DNL2*DNR2/(1.0+RNR*SNRN+RNK*SNKN)                              DGEOM   68
      DNR=R2*DNR2+DN*SNRN                                               DGEOM   69
      DNK=Z2*DNR2+DN*SNKN                                               DGEOM   70
      IF(IS .EQ. 1)CS111=(R*SN*(RMSN-0.5*RLRMSN)                        DGEOM   71
      DA22=DR2*(R2-0.5*DR2)+DZ2*(Z2-0.5*DZ2)                            DGEOM   72
      IF(IS .EQ. 1)DR11=-(DRMSN*RNR+RMSN*DNR)                           DGEOM   73
      DR22=RNR*(R22+RNK*DZ22+DNR*R22+DNK*Z22                            DGEOM   74
      DO 89 K=1,LAYER                                                   DGEOM   75
      DEPS1(K)=DA11-ZETA(K)*DR11                                        DGEOM   76
      DEPS2(K)=DA22-ZETA(K)*DR22                                        DGEOM   77
      IF(IP .GT. 0) CALL RMSTRS(N,K)                                    DGEOM   78
      IF(IP .LE. 0)CALL STRESS(N,K)                                     DGEOM   79
C     CODING GO TO 100 STOPS BEST ESTIMATE OF SURFACE STRAINS.          DGEOM   80
   89 CONTINUE                                                          DGEOM   81
      IF(IP .EQ. 2)GOTO 100                                             DGEOM   82
      DA1(N) = DA1(N) + DA11                                            DGEOM   83
      DR1(N) = DR1(N) + DR11                                            DGEOM   84
      DR2(N) = DR2(N) + DR22                                            DGEOM   85
      EPSL1(N) = EPSL1(N) + DA11 - ZL*DR11                              DGEOM   86
      EPSL2(N)=EPSL2(N)     -ZL*DR22                                    DGEOM   87
      EPSU1(N) = EPSU1(N) + DA11 - ZU*DR11                              DGEOM   88
      EPSU2(N)=EPSU2(N)     -ZU*DR22                                    DGEOM   89
      GOTO 100                                                          DGEOM   90
```

```
  189 CONTINUE
      IF(N .EQ. N1A) DA22R = DA22
      DA2(N) = DA2 (N) + DA22
      DA22R=0.5*(DA22R+DA22)
      EPSL2(N)=EPSL2(N)+DA22R
      EPSU2(N)=EPSL2(N)+DA22R
      DA22R=DA22
      CALL RESULT(N)
      IF(N.EQ.N2A-1) GOTO 91
      GOTO 90
  190 CONTINUE
      CALL RESULT(N)
   90 CONTINUE
   91 CONTINUE
      N=N2A
      EPSL2(N)=EPSL2(N)+DA22P
      EPSU2(N)=EPSL2(N)+DA22R
      CALL SYMTRY
      IF(NCYCLE.EQ.0)GOTO 820
      STREN=CR*STREN
      IF(IFCE1 .EQ. 2)STREN=2.0*STREN
      IF(IFCE2 .EQ. 2) STREN=2.0*STREN
      IF(NCYCLE.NE.NCYNLP(NLP))GOTO 820
C         WRITE LMATRIX
      NLP=NLP+1
      WRITE(6,811)NCYCLE,TIME, (N,N=N1A,N2A)
      DO 802 K=1,LAYER
      WRITE(6,813)K,(LMAT(N,K),N=N1A,N2A)
  802 WRITE(6,813)K,(LMATM(N,K),N=N1A,N2A)
  820 IF(NCYCLE .NE. NCYCH(NNN))GOTO 700
C        WRITE SURFACE NORMAL VECTOR
      IF(JCHK(3)) 700,700,800
  800 WRITE(6,900) NCYCLE,TIME
      WRITE(6,910)
      WRITE(6,920)(N,SNR(N),SNK(N),SNRM(N),SNKM(N),N=N1A,N2A)
C          TEMPORARY PRINT OF SURFACE STRAINS
      WRITE(6,770)
      DO 71 N=N1A,N2A
      EPCU2=SQRT(1.0+2.0*EPSU2(  N)*GU22(N))-1.0
      EPCU1=SQRT(1.0+2.0*EPSU1(  N)*GU11(N))-1.0
      EPCL2=SQRT(1.0+2.0*EPSL2(  N)*GL22(N))-1.0
      EPCL1=SQRT(1.0+2.0*EPSL1(  N)*GL11(N))-1.0
      WRITE(6,771)N,EPCU1,EPCU2,EPSU1(N),EPSU2(N)
     1                 ,EPCL1,EPCL2,EPSL1(N),EPSL2(N)
   71 CONTINUE
      WRITE(6,770)
  770 FORMAT(//' SURFACE STRAINS, UPPER/LOWER,  (EC1,EC2,E11,E22)',//)
  771 FORMAT(I5,1P4E20.13,/,5X,4E20.13)
  700 RETURN
C
  811 FORMAT(10H0TIME STEP,I5,6X,4HTIME,1PE16.7,1P,,10X,
     1 40HSUBDIVISIONS OF TIME INCREMENT IN STRESS///20X,
     2 9HLMAT(N,K)//9H    K    N=,40I3/(8X,40I3))
  813 FORMAT(/I4,5X,40I3/(9X,40I3))
  900 FORMAT(10H0TIME STEP,I5,6X,5H TIME,F16.8/)
  910 FORMAT(13X,32HSURFACE NORMAL VECTOR COMPONENTS //
     1 3X,1HN,11X,6HSNR(N),19X,6HSNK(N),19X,7HSNRM(N),18X,7HSNKM(N))
  920 FORMAT((I4,4(2X,E23.16)))
      END
      SUBROUTINE GRAD(ND)
C                    *CALL MAIN HERE
C         IDCS1=1,2,OR3,  IDCS2=1,2,3,OR4,   2/28/76   1/7/76
C         IR=1 DENOTES MESH POINTS,  IR=2 DENOTES MESHES
```

```
C        DIFFERENCES FOR RV CENTERED AT MESH POINTS.  FOR RN IN MESHES   GRAD      6
      N=ND                                                               GRAD      7
      IF(ID .EQ. 2)GOTO 10                                               GRAD      8
      IF(N .LT. NN) GOTO 5                                               GRAD      9
C        SPECIAL CODING FOR N=NN (A FREE END)                            GRAD     10
       R2=RTD2*( R(N-2)-4.0* R(N-1) +3.0* R(N))                          GRAD     11
       Z2=RTD2*( Z(N-2)-4.0* Z(N-1) +3.0* Z(N))                          GRAD     12
      DR2=RTD2*(DR(N-2)-4.0*DR(N-1) +3.0*DR(N))                          GRAD     13
      DZ2=RTD2*(DZ(N-2)-4.0*DZ(N-1) +3.0*DZ(N))                          GRAD     14
       R22=RD22*(2.0* R(N)-5.0* R(N-1)+4.0* R(N-2)- R(N-3))              GRAD     15
       Z22=RD22*(2.0* Z(N)-5.0* Z(N-1)+4.0* Z(N-2)- Z(N-3))              GRAD     16
      DR22=RD22*(2.0*DR(N)-5.0*DR(N-1)+4.0*DR(N-2)-DR(N-3))              GRAD     17
      DZ22=RD22*(2.0*DZ(N)-5.0*DZ(N-1)+4.0*DZ(N-2)-DZ(N-3))              GRAD     18
      GOTO 20                                                            GRAD     19
    5 CONTINUE                                                           GRAD     20
       R2=RTD2*( R(N+1)- R(N-1))                                         GRAD     21
       Z2=RTD2*( Z(N+1)- Z(N-1))                                         GRAD     22
      DR2=RTD2*(DR(N+1)-DR(N-1))                                         GRAD     23
      DZ2=RTD2*(DZ(N+1)-DZ(N-1))                                         GRAD     24
       R22=RD22*( R(N-1)-2.0* R(N)+ R(N+1))                              GRAD     25
       Z22=RD22*( Z(N-1)-2.0* Z(N)+ Z(N+1))                              GRAD     26
      DR22=RD22*(DR(N-1)-2.0*DR(N)+DR(N+1))                              GRAD     27
      DZ22=RD22*(DZ(N-1)-2.0*DZ(N)+DZ(N+1))                              GRAD     28
      GOTO 20                                                            GRAD     29
   10  R2=RTD2M*( R(N+1)- R(N))                                          GRAD     30
       Z2=RTD2M*( Z(N+1)- Z(N))                                          GRAD     31
      DR2=RTD2M*(DR(N+1)-DR(N))                                          GRAD     32
      DZ2=RTD2M*(DZ(N+1)-DZ(N))                                          GRAD     33
      IF(N .LT. NN-1) GOTO 15                                            GRAD     34
C        SPECIAL CODING FOR N=NN (A FREE END)                            GRAD     35
       R22=RD22M*(3.0* R(N+1)-7.0* R(N)+5.0* R(N-1)- R(N-2))             GRAD     36
       Z22=RD22M*(3.0* Z(N+1)-7.0* Z(N)+5.0* Z(N-1)- Z(N-2))             GRAD     37
      DR22=RD22M*(3.0*DR(N+1)-7.0*DR(N)+5.0*DR(N-1)-DR(N-2))             GRAD     38
      DZ22=RD22M*(3.0*DZ(N+1)-7.0*DZ(N)+5.0*DZ(N-1)-DZ(N-2))             GRAD     39
      GOTO 20                                                            GRAD     40
   15 CONTINUE                                                           GRAD     41
       R22=RD22M*( R(N-1)- R(N)- R(N+1)+ R(N+2))                         GRAD     42
       Z22=RD22M*( Z(N-1)- Z(N)- Z(N+1)+ Z(N+2))                         GRAD     43
      DR22=RD22M*(DR(N-1)-DR(N)-DR(N+1)+DR(N+2))                         GRAD     44
      DZ22=RD22M*(DZ(N-1)-DZ(N)-DZ(N+1)+DZ(N+2))                         GRAD     45
   20 RETURN                                                             GRAD     46
      END                                                                GRAD     47
      SUBROUTINE STRESS(ND,KD)                                           STRESS    2
                                                                         STRESS    3
C        CONSTITUTIVE RELATION--LINEARLY PLASTIC, OR PLASTIC-(=PERFECTLY STRESS    4
C        PLASTIC OR =STRAIN HARDENING).  OPTIONAL STRAIN RATE DEPENDENCE STRESS    5
      N=ND                                                               STRESS    6
      K=KD                                                               STRESS    7
      SS11=0.0                                                           STRESS    8
      SS22=0.0                                                           STRESS    9
      LMNK=0                                                             STRESS   10
      KN=(K-1)*NSEL                                                      STRESS   11
      ZETAK=2.0*ZETA(K)                                                  STRESS   12
      G11 =A11-ZETAK*H11                                                 STRESS   13
      G22 =A22-ZETAK*H22                                                 STRESS   14
      DG=G11*G22                                                         STRESS   15
      SRG=SQRT(DG)                                                       STRESS   16
      RG=1.0/DG                                                          STRESS   17
      GH11 = RG * G22                                                    STRESS   18
      GH22 = RG * G11                                                    STRESS   19
      DEPS11=GH11*DEPS1(K)                                               STRESS   20
      DEPS22=GH22*DEPS2(K)                                               STRESS   21
      DSIG1 = GH11*(DEPS1(K) + GH12*DEPS2(K))                            STRESS   22
```

```
      DSIG22 = ENAT*(DEPS22 + ENU*DEPS11)                                STRESS   23
      IF(ISW.GT.0)EPSDOT=SQRT(DEPS11*(DEPS11-DEPS22)+
     1  DEPS22**2)/DELTAT                                               STRESS   25
      DO 403 J=1,NSEL                                                  STRESS   26
      KJ=KK+J                                                          STRESS   27
C        IP= DENOTES MESH POINTS.  IP=2 DENOTES MESHES                  STRESS   28
      IF(IP .EQ. 2)GOTO 22                                             STRESS   29
      SIG11I=G11*SIG1(N,KJ)                                            STRESS   30
      SIG22I=G22*SIG2(N,KJ)                                            STRESS   31
      GOTO 23                                                          STRESS   32
   22 SIG11I=G11*SIG1M(N,KJ)                                           STRESS   33
      SIG22I=G22*SIG2M(N,KJ)                                           STRESS   34
   23 CONTINUE                                                         STRESS   35
      SIGYSQ=SIGZSQ(J)                                                 STRESS   36
      IF(ISP.GT.0)SIGYSQ=SIGYSQ*(1.0+(EPSOUT/DSR(J))**PSR(J))**2       STRESS   37
      L=1                                                              STRESS   38
      LC=1                                                             STRESS   39
      SIG11L=SIG11I+DSIG11                                             STRESS   40
      SIG22L=SIG22I+DSIG22                                             STRESS   41
      IF(ISW.LT.0)GOTO 2                                               STRESS   42
      PHIT=SIG11L*(SIG11L-SIG22L)+SIG22L**2-SIGYSQ                     STRESS   43
      IF(PHIT.LE.0.0)GOTO 2                                            STRESS   44
      L=INT(YLDFAC*(SQRT((PHIT+SIGYSQ)/SIGYSQ)-1.0))+1                 STRESS   45
  100 SIG11=SIG11I                                                     STRESS   46
      SIG22=SIG22I                                                     STRESS   47
      IF(L.EQ.1)GOTO 3                                                 STRESS   48
      LC=1                                                             STRESS   49
      FLOATL=1.0/FLOAT(L)                                              STRESS   50
      DSG11L = DSIG11*FLOATL                                           STRESS   51
      DSG22L = DSIG22*FLOATL                                           STRESS   52
  101 SIG11L= SIG11+DSG11L                                             STRESS   53
      SIG22L= SIG22+DSG22L                                             STRESS   54
      PHIT=SIG11L*(SIG11L-SIG22L)+SIG22L**2-SIGYSQ                     STRESS   55
      IF(PHIT .GT. 0.0)GOTO 3                                          STRESS   56
C     PLASTIC                                                          STRESS   57
    2 SIG11 = SIG11L                                                   STRESS   58
      SIG22 = SIG22L                                                   STRESS   59
      GO TO 9                                                          STRESS   60
C     PLASTIC                                                          STRESS   61
    3 SIG11D = (2.0-ENU)*SIG11-(1.0-2.0*ENU)*SIG22                     STRESS   62
      SIG22D = (2.0-ENU)*SIG22-(1.0-2.0*ENU)*SIG11                     STRESS   63
      AA=SIG11L*(SIG11D-SIG22D)+SIG22D**2                              STRESS   64
      BB=-(SIG11L*(2.0*SIG11D-SIG22D)+SIG22L*(2.0*SIG22D-SIG11L))      STRESS   65
      D=BB**2-4.*AA*PHIT                                               STRESS   66
      IF(L .GT. LMAX) LMAX=L                                           STRESS   67
      IF(AA)8,16,4                                                     STRESS   68
    8 WRITE(6,10)                                                      STRESS   69
   10 FORMAT(1H ,4X,14HAA NEGATIVE AT)                                 STRESS   70
      GOTO 12                                                          STRESS   71
    4 IF(D .LT. 0.0 .OR. B .GT. 0.0)GOTO 16                            STRESS   72
      TAMBDA=(-B-SQRT(D))/(2.*AA)                                      STRESS   73
      SIG11 = SIG11L-TAMBDA*SIG11D                                     STRESS   74
      SIG22 = SIG22L-TAMBDA*SIG22D                                     STRESS   75
    9 LC=LC+1                                                          STRESS   76
      IF(LC-L)101,101,102                                              STRESS   77
   16 L=L+1                                                            STRESS   78
      IF(L .LE. 100) GOTO 100                                          STRESS   79
      WRITE(6,104)                                                     STRESS   80
  104 FORMAT(1H ,4X,36HSTRESS CALCULATION UNSATISFACTORY AT)           STRESS   81
   12 WRITE(6,105)NCYCLE,N,K,J,L,LC                                    STRESS   82
  105 FORMAT(10X,9HTIME STEP=I4,5X,2HN=I3,5X,2HK=I2,5X,2HJ=I3,5X,2HL=I3,  STRESS   83
     1  5X,2HLC=I3)                                                    STRESS   84
      WRITE(6,106)EPSL1(N),EPSL2(N),EPS01(N),EPS02(N),DEPS1(N),        STRESS   85
```

```
     1DEPS2(K),SIG1(N,KJ),SIG2(N,KJ),DSIG11,DSIG22,SIG11,SIG22,          STRESS   86
     2SIG11L,SIG22L,SIG11C,SIG22C),AA,R,PHIT                            STRESS   87
  104 FORMAT(10X,7HEPSL1 =E15.8,3X,7HEPSL2 =E15.8,3X,7HEPSU1 =E15.8,     STRESS   88
     1 3X,7HEPSU2 =E15.8/10X,7HDEPS1 =E15.8,3X,7HDEPS2 =E15.8,3X,       STRESS   89
     2 7H SIG1 =E15.8,3X,7H SIG2 =E15.8/10X,7HDSIG11=E15.8,3X,          STRESS   90
     3 7HDSIG22=E15.8,3X,7HSIG11 =E15.8,3X,7HSIG22 =E15.8/10X,          STRESS   91
     4 7HSIG11L=E15.8,3X,7HSIG22L=E15.8,3X,7HSIG11C=E15.8,3X,           STRESS   92
     5 7HSIG22L=E15.8/13X,4HAA =E15.8,3X,7HR =E15.8,3X,6HPHIT =E15.8)   STRESS   93
       NC3DP(NN3D)=NCYCLE                                               STRESS   94
       CALL PDATA (2)                                                   STRESS   95
       CALL PDATA (3)                                                   STRESS   96
       CALL PDATA (4)                                                   STRESS   97
       STOP' STRESS'                                                    STRESS   98
  102 SS11=SS11+SIG11*WT(J)                                             STRESS   99
       SS22=SS22+SIG22*WT(J)                                            STRESS  100
       IF(IP .EQ. 2)GOTO 122                                            STRESS  101
C         IP=1 DENOTES MESH POINTS.  IP=2 DENOTES MESHES                STRESS  102
       SIG1(N,KJ)=GR11*SIG11                                            STRESS  103
       SIG2(N,KJ)=GR22*SIG22                                            STRESS  104
       GOTO 803                                                         STRESS  105
  122 SIG1M(N,KJ)=GR11*SIG11                                            STRESS  106
       SIG2M(N,KJ)=GR22*SIG22                                           STRESS  107
  803 CONTINUE                                                          STRESS  108
       IF(N.EQ.N1M.OR.N.EQ.N2M)SHG=0.5*SHG                              STRESS  109
       IF(IGAUSS .EQ. 1) SHG = W(K)*SHG                                 STRESS  110
       IF(IP.EQ.1)STHEN=STHEN+((SS11+SS22)**2-(1.+ENU)*2.*SS11*SS22)*SHG STRESS  111
       SS1MN(N)=GR11*SS11                                               STRESS  112
       SS2MN(K)=GR22*SS22                                               STRESS  113
       IF(IP .EQ. 1)LMAT(N,K)=LMNK                                      STRESS  114
       IF(IP .EQ. 2)LMATM(N,K)=LMNK                                     STRESS  115
       RETURN                                                           STRESS  116
       END                                                              STRESS  117
       SUBROUTINE HMSTRS(ND,KD)                                         HMSTRS    2
C                    *CALL MAIN HERE                                    TAF       1
C          STRESS FOR BEAMS                                            HMSTRS    4
C       CONSTITUTIVE RELATION--LINEARLY ELASTIC, OR ELASTIC-(=PERFECTLY HMSTRS    5
C       PLASTIC OR =STRAIN HARDENING).  OPTIONAL STRAIN RATE DEPENDENCE HMSTRS    6
       N=ND                                                             HMSTRS    7
       K=KD                                                             HMSTRS    8
       SS22=0.0                                                         HMSTRS    9
       LMNK=0                                                           HMSTRS   10
       L=0                                                              HMSTRS   11
       KN=(K-1)*NSEL                                                    HMSTRS   12
       G22 = A22+2.*ZETA(K)*EE2                                         HMSTRS   13
       IF(IP .EQ. 1)SHG = SQRT(G22)                                     HMSTRS   14
       GR22 = 1.0/G22                                                   HMSTRS   15
       DEPS22=GR22*DEPS2(K)                                             HMSTRS   16
       DSIG22=E*DEPS22                                                  HMSTRS   17
       DO 803 J=1,NSEL                                                  HMSTRS   18
       KJ=KN+J                                                          HMSTRS   19
C         IP=1 DENOTES MESH POINTS.  IP=2 DENOTES MESHES               HMSTRS   20
       IF(IP .EQ. 1)SIG22I=G22*SIG2(N,KJ)                               HMSTRS   21
       IF(IP .EQ. 2)SIG22I=G22*SIG2M(N,KJ)                              HMSTRS   22
       SIGYZ=SIGZZ(J)                                                   HMSTRS   23
       IF(ISR.GT.0)SIGYZ=SIGYZ*(1.+(ABS(DEPS22)/(DELTAT*DSR(J)))**PSR(J)) HMSTRS   24
       SIG22=SIG22I + DSIG22                                            HMSTRS   25
       IF(ISR.LT.0)GOTO 102                                             HMSTRS   26
       IF(ABS(SIG22).GT.SIGYZ)L=1+IMT(4.0*((ABS(SIG22)-SIGYZ)/SIGYZ))   HMSTRS   27
       IF(L.GT.LMNK)LMNK=L                                              HMSTRS   28
       IF(SIG22 .LT. -SIGYZ) SIG22=-SIGYZ                               HMSTRS   29
       IF(SIG22 .GT. SIGYZ) SIG22 = SIGYZ                               HMSTRS   30
  102 SS22=SS22+SIG22*WT(J)                                             HMSTRS   31
       IF(IP .EQ. 1)SIG2(N,KJ)=GR22*SIG22                               HMSTRS   32
```

```
      IF(IR .EQ. 2)SIG2M(N,KJ)=GR22*SIG22                              RFSTRS    33
  803 CONTINUE                                                         RFSTRS    34
      IF(N.EQ.N1R.OR.N.EQ.N2R)SHG=0.5*SHG                              RFSTRS    35
      IF(IP.EQ.1)STRENE=STRENE+SHG*SS22**2*A(K)                        RFSTRS    36
      SS2MN(K)=GR22*SS22                                               RFSTRS    37
      IF(IP .EQ. 1)LMAT(N,K)=LMNK                                      RFSTRS    38
      IF(IP .EQ. 2)LMATM(N,K)=LMNK                                     RFSTRS    39
      RETURN                                                           RFSTRS    40
      END                                                              RFSTRS    41
      SUBROUTINE RESULT(ND)                                            RESULT     2
C                    *CALL MAIN HERE                                   TAR        1
C     THE CALCULATION OF THE STRESS AND MOMENT RESULTANTS FOLLOW       RESULT     4
C        IRCE1=1,2,OR3.   IRCE2=1,2,3,OR4.   8/28/74   1/7/76          RESULT     5
      N=ND                                                             RESULT     6
      TR=TA*SRA                                                        RESULT     7
      IF(IR .GT. 0)GOTO 7                                              RESULT     8
      IF(IGAUSS .EQ. 1)GOTO 9                                          RESULT     9
      SUMA11 = SS1MN(1)                                                RESULT    10
      SUMA22 = SS2MN(1)                                                RESULT    11
      SUMB11 = SS1MN(1)*ZETA(1)                                        RESULT    12
      SUMB22 = SS2MN(1)*ZETA(1)                                        RESULT    13
      SUMC11 = SS1MN(1)*ZETASQ(1)                                      RESULT    14
      SUMC22 = SS2MN(1)*ZETASQ(1)                                      RESULT    15
      IF(LAYER .EQ. 1)GOTO 5                                           RESULT    16
      DO 4 K=2,LAYER                                                   RESULT    17
      SUMA11 = SUMA11 + SS1MN(K)                                       RESULT    18
      SUMA22 = SUMA22 + SS2MN(K)                                       RESULT    19
      SUMB11 = SUMB11 + SS1MN(K)*ZETA(K)                               RESULT    20
      SUMB22 = SUMB22 + SS2MN(K)*ZETA(K)                               RESULT    21
      SUMC11 = SUMC11 + SS1MN(K)*ZETASQ(K)                             RESULT    22
      SUMC22 = SUMC22 + SS2MN(K)*ZETASQ(K)                             RESULT    23
    4 CONTINUE                                                         RESULT    24
    5 CONTINUE                                                         RESULT    25
      Q11=SUMA11-RT*SUMB11                                             RESULT    26
      Q22=SUMA22-RT*SUMB22                                             RESULT    27
      F11=SUMB11-(RT+RM11)*SUMC11                                      RESULT    28
      F22=SUMB22-(RT+RM22)*SUMC22                                      RESULT    29
      IF(IP .EQ. 2)GOTO 17                                             RESULT    30
C        IP=1 DENOTES MESH POINTS.  IP=2 DENOTES MESHES                RESULT    31
C        DIFFERENCES FOR FM CENTERED AT MESH POINTS.  FOR FN IN MESHES RESULT    32
      IF(N .EQ. N1R .AND. IRCE1 .EQ. 3)F22=0.0                         RESULT    33
      IF(N .EQ. N2R .AND. IRCE2 .EQ. 3)F22=0.0                         RESULT    34
      IF(N .NE. NN)GOTO 16                                             RESULT    35
C        N=NN.  ASSUME IRCE2 =4.  COMPUTE ALL RESULTANTS.              RESULT    36
      FM11R(N)=TR*F11*SNR(N)                                           RESULT    37
      FM22R(N)=0.0                                                     RESULT    38
      FM22K(N)=0.0                                                     RESULT    39
      FN1TR(N)=TR*Q11*R(N)                                             RESULT    40
C        AT FREE END.  FN2J(N)= (-DM22/DETA)*NJ                        RESULT    41
      IF(SNR(N-1) .NE. 0.0)F22=FM22R(N-1)/SNR(N-1)                     RESULT    42
      IF(SNR(N-1) .EQ. 0.0)F22=FM22K(N-1)/SNK(N-1)                     RESULT    43
      IF(SNR(N-2) .NE. 0.0)F11=FM22R(N-2)/SNR(N-2)                     RESULT    44
      IF(SNR(N-2) .EQ. 0.0)F11=FM22K(N-2)/SNK(N-2)                     RESULT    45
      F22=RTD2*(4.0*F22-F11)                                           RESULT    46
      FN2R(N)=F22*SNR(N)                                               RESULT    47
      FN2K(N)=F22*SNK(N)                                               RESULT    48
      GOTO 18                                                          RESULT    49
   16 CONTINUE                                                         RESULT    50
      FM11R(N)=TR*F11*SNR(N)                                           RESULT    51
C     FM11K(N)=TR*F11*SNK(N)   IS NOT USED                             RESULT    52
      FM22R(N)=TR*F22*SNR(N)                                           RESULT    53
      FM22K(N)=TR*F22*SNK(N)                                           RESULT    54
      GOTO 18                                                          RESULT    55
```

```
   17 CONTINUE                                                          RESULT     56
      CS**2=CS211*F11+CS222*F22                                         RESULT     57
      FN1TH(N)=TH*G11*0.5*(F(N)+F(N+1))                                 RESULT     58
C        FN1TH(N) AND FN11H(N) ARE NOT USED IF IS=2.                    RESULT     59
      FN2R(N)=TH*(C22*H2+SNRM(N)*CSM2)                                  RESULT     60
      FN2K(N)=TH*(C22*Z2+SNKM(N)*CSM2)                                  RESULT     61
   18 CONTINUE                                                          RESULT     62
      RETURN                                                            RESULT     63
    7 SUMA22=0.0                                                        RESULT     64
      SUMH22=0.0                                                        RESULT     65
      SUMC22=0.0                                                        RESULT     66
      DO 8 K=1,LAYER                                                    RESULT     67
      SUMA22 = SUMA22+W(K)*SS2MN(K)                                     RESULT     68
      SUMH22 = SUMH22 + SS2MA(K)*WZETA(K)                               RESULT     69
      SUMC22 = SUMC22 + SS2MA(K)*WZETSQ(K)                              RESULT     70
    8 CONTINUE                                                          RESULT     71
      GOTO 5                                                            RESULT     72
    9 SUMA11 = 0.0                                                      RESULT     73
      SUMH11 = 0.0                                                      RESULT     74
      SUMC11 = 0.0                                                      RESULT     75
      DO 10 K=1,LAYER                                                   RESULT     76
      SUMA11 = SUMA11 + SS1MA(K)*W(K)                                   RESULT     77
      SUMH11 = SUMH11 + SS1MA(K)*WZETA(K)                               RESULT     78
      SUMC11 = SUMC11 + SS1MA(K)*WZETSQ(K)                              RESULT     79
   10 CONTINUE                                                          RESULT     80
      GOTO 7                                                            RESULT     81
      END                                                               RESULT     82
      SUBROUTINE MOTION                                                 MOTION      2
C                    *CALL MAIN HERE                                    TAB         1
C                                                                       MOTION      4
      IF(LOAD .EQ. 0)GOTO 30                                            MOTION      5
      ENS=0.0                                                           MOTION      6
      CALL PROBK                                                        MOTION      7
C                                                                       MOTION      8
C        TRCE1=1,2,OR3.  TRCE2=1,2,3,OR4.  8/28/74  1/7/76              MOTION      9
   30 DO 130 N=N1V,N2V                                                  MOTION     10
      IF(N .NE. NN)GOTO 31                                              MOTION     11
C        SPECIAL CODING FOR N=NN (A FREE END)                           MOTION     12
      VR=RD22*(2.0*FM22R(N)-5.0*FM22R(N-1)+4.0*FM22R(N-2)-FM22R(N-3))   MOTION     13
     1  +2.0*RTD2M*(FN2R(N)-FN2R(N-1))                                  MOTION     14
      VZ=RD22*(2.0*FM22K(N)-5.0*FM22K(N-1)+4.0*FM22K(N-2)-FM22K(N-3))   MOTION     15
     1  +2.0*RTD2M*(FN2K(N)-FN2K(N-1))                                  MOTION     16
      IF(IS .EQ. 1)VR=VR-FN11H(N)-FN1TH(N)                              MOTION     17
      GOTO 35                                                           MOTION     18
   31 CONTINUE                                                          MOTION     19
C        DIFFERENCES FOR FN (ENTERED AT MESH POINTS,  FOR FN IN MESHES  MOTION     20
      VR=RD22*(FM22R(N-1)-2.0*FM22R(N)+FM22R(N+1))                      MOTION     21
     1 +RTD2M*(FN2R(N)-FN2R(N-1))                                       MOTION     22
      VZ=RD22*(FM22K(N-1)-2.0*FM22K(N)+FM22K(N+1))                      MOTION     23
     1 +RTD2M*(FN2K(N)-FN2K(N-1))                                       MOTION     24
      IF(IS .EQ. 1)VR=VR-FN11H(N)-0.5*(FN1TH(N-1)+FN1TH(N))             MOTION     25
   35 CONTINUE                                                          MOTION     26
      IF(LOAD.EQ.0)GOTO 50                                              MOTION     27
      VR=VR-SNR(N)*P(N)                                                 MOTION     28
      VZ=VZ-SNK(N)*P(N)                                                 MOTION     29
   50 DRI=DR(N)                                                         MOTION     30
      DZI=DZ(N)                                                         MOTION     31
      DRS=DRI+VR*TEMP(N)                                                MOTION     32
      DZS=DZI+VZ*TEMP(N)                                                MOTION     33
      IF(TEMP.EQ.0.0)GOTO 115                                           MOTION     34
C                VISCOUS DAMPING                                        MOTION     35
      DRS=DRS-(DRS+DRI)*C1                                              MOTION     36
      DZS=DZS-(DZS+DZI)*C1                                              MOTION     37
```

```
  115 DR(N)=DRS                                                          MOTION     38
      DZ(N)=DZS                                                          MOTION     39
  130 CONTINUE                                                           MOTION     40
      CALL RELAX                                                         MOTION     41
      CALL KINET                                                         MOTION     42
      IF(LOAD) 65,75,65                                                  MOTION     43
   65 CALL PSTRN                                                         MOTION     44
      EN=0.5*(A*(EPS+ENP)                                                MOTION     45
      ENR=EPS                                                            MOTION     46
      IF(IRCE1 .EQ. 2)EN=2.0*EN                                          MOTION     47
      IF(IRCE2 .EQ. 2)EN=2.0*EN                                          MOTION     48
      TNRG=TNRG+EN                                                       MOTION     49
   75 PLAST=TNRG-CINET-STREN-TDAMP                                       MOTION     50
C                                                                        MOTION     51
      IF(NCYCLE .NE. NCYCP(NNN)) GOTO 140                                MOTION     52
      WRITE(6,99991) NCYCLE,TIME,CINET,STREN,PLAST,TNRG                  MOTION     53
      NNN=NNN+1                                                          MOTION     54
  140 RETURN                                                             MOTION     55
C                                                                        MOTION     56
99991 FORMAT(//10H TIME STEP,I5,3X,5HTIME=,F16.8,3X,4HKINETIC=,E15.8,3X, MOTION     57
     1HPLASTIC=,E15.8,3X,4HPLASTIC=,E15.8/14H TOTAL ENERGY=,E15.8)        MOTION     58
      END                                                                MOTION     59
      SUBROUTINE WRTAPE(KEY)                                             WRTAPE      2
C        WRTAPE HAS BEEN DELETED.                                        WRTAPE      3
      STOP* WRTAPE DELETED*                                              WRTAPE      4
      END                                                                WRTAPE      5
      SUBROUTINE STRAIN                                                  STRAIN      2
C                    *CALL MAIN HERE                                     T&F         1
C                                                                        STRAIN      4
C     PRINT STRAINS ON INNER OR OUTER FACES                              STRAIN      5
C                                                                        STRAIN      6
      DIMENSION EPSANG(6),EPSANG(6)                                      STRAIN      7
C        ARRAYS FOR EXTREME STRAINS ---  IN STRAIN                       STRAIN      8
      DIMENSION EPC(4),EPSXT(8),NEPSXT(8),LEPSX(8)                       STRAIN      9
      DATA(LEPSX(L),L=1,8)/8HEPSU1MAX,8HEPSU2MAX,8HEPSL1MAX,8HEPSL2MAX,  STRAIN     10
     1                     8HEPSU1MIN,8HEPSU2MIN,8HEPSL1MIN,8HEPSL2MIN/  STRAIN     11
      DATA PI/3.14159265358979/                                          STRAIN     12
      IF(NCYCLE .GT. 0)GOTO 25                                           STRAIN     13
C     ---------------------- INITIAL ENTRY ----------------------------- STRAIN     14
      DO 20 I=1,NSTRN                                                    STRAIN     15
      J1=NT1(I)                                                          STRAIN     16
      J2=NT2(I)                                                          STRAIN     17
      J3=NT3(I)                                                          STRAIN     18
      A11 = DA1(J1) + DN1(I)*(DB1(J2)-DA1(J1))                           STRAIN     19
      R11 = DR1(J1) + DN1(I)*(DB1(J2)-DR1(J1))                           STRAIN     20
      R22 = DR2(J1) + DN1(I)*(DR2(J2)-DR2(J1))                           STRAIN     21
      A22 = DA2(J3) + DN3(I)*(DA2(J3+1)-DA2(J3))                         STRAIN     22
      ZR = ZL                                                            STRAIN     23
      IF(MTAG(I) .EQ. 1)ZR=ZL                                            STRAIN     24
      G11 = A11 - 2.0*ZR*R11                                             STRAIN     25
      G22 = A22 - 2.0*ZR*R22                                             STRAIN     26
   15 G111(I)=1.0/G11                                                    STRAIN     27
      G122(I)=1.0/G22                                                    STRAIN     28
      ANGEL=ANGLE(I)*PI/180.0                                            STRAIN     29
      SA=SIN(ANGEL)                                                      STRAIN     30
      SR=COS(ANGEL)                                                      STRAIN     31
      ASA(I)=2.0*SA**2                                                   STRAIN     32
      RSA(I)=2.0*SR**2                                                   STRAIN     33
      ANGEL=ANGLR(I)*PI/180.0                                            STRAIN     34
      SA=SIN(ANGEL)                                                      STRAIN     35
      SR=COS(ANGEL)                                                      STRAIN     36
      ASR(I)=2.0*SA**2                                                   STRAIN     37
      RSR(I)=2.0*SR**2                                                   STRAIN     38
```

```
   20 CONTINUE                                                          STRAIN     39
      GOTO 71                                                           STRAIN     40
C     ------------------------------------------------------------------ STRAIN     41
   25 LINK=1                                                            STRAIN     42
C         FIND EXTREMES OF STRAIN AND MAXIMUM DEFLECTION                STRAIN     43
      DO 901 L=1,4                                                      STRAIN     44
      EPSXT(L)=-100000.                                                 STRAIN     45
  901 EPSXT(L+4)= 100000.                                               STRAIN     46
      DEFXT=0.0                                                         STRAIN     47
      DO 905 N=N1M,N2M                                                  STRAIN     48
      EPC(1)=EPSU1(N)*GU11(N)$  EPC(2) = EPSL2(N)*GU22(N)               STRAIN     49
      EPC(3)=EPSL1(N)*GL11(N)$  EPC(4)=EPSL2(N)*GL22(N)                 STRAIN     50
      DO 905 L=1,8                                                      STRAIN     51
      IF(L .GT. 4)GOTO 902                                              STRAIN     52
      IF(EPSXT(L) .GE. EPC(L))GOTO 905                                  STRAIN     53
      EPSXT(L)=EPC(L)$  NEPSXT(L)=N$  GOTO 905                          STRAIN     54
  902 IF(EPSXT(L) .LE. EPC(L-4))GOTO 905                                STRAIN     55
      EPSXT(L)=EPC(L-4)$     NEPSXT(L)=N                                STRAIN     56
  905 CONTINUE                                                          STRAIN     57
      DO 910 L=1,8                                                      STRAIN     58
      EPSXT(L)=SQRT(1.0+2.0*EPSXT(L))-1.0                               STRAIN     59
      IF(L .GT. 4)GOTO 906                                              STRAIN     60
      IF(EPSX(L) .GE. EPSXT(L))GOTO 910                                 STRAIN     61
      GOTO 907                                                          STRAIN     62
  906 IF(EPSX(L) .LE.EPSXT(L))GOTO 910                                  STRAIN     63
  907 EPSX(L)=EPSXT(L)$  TEPSX(L)=TIME                                  STRAIN     64
       NEPSX(L)=NEPSXT(L)                                               STRAIN     65
  910 CONTINUE                                                          STRAIN     66
      DO 912 N=N1V,N2V                                                  STRAIN     67
      AY3MA=(PZ(N)-Z(N))**2 + (ZZ(N)-Z(N))**2                           STRAIN     68
      IF(DEEXT .GE. AY3MA)GOTO 912                                      STRAIN     69
      DEEXT=AY3MA$   NDEEXT=N                                           STRAIN     70
  912 CONTINUE                                                          STRAIN     71
      DEEXT = SQRT(DEEXT)                                               STRAIN     72
      IF(DEEX .GE. DEEXT) GOTO 913                                      STRAIN     73
      DEEX=DEEXT$   NDEEX=NDEEXT$  TDEEX=TIME                           STRAIN     74
  913 CONTINUE                                                          STRAIN     75
C         CHECK FOR SURFACE STRAIN PRINT                                STRAIN     76
      IF(NCYCLE .LT. NPRINT)GOTO 40                                     STRAIN     77
   30 NPRINT=NPRINT+NDELP                                               STRAIN     78
      LINK=2                                                            STRAIN     79
   40 DO 46 I=1,NSTRN                                                   STRAIN     80
      J1=N1(I)                                                          STRAIN     81
      J2=N2(I)                                                          STRAIN     82
      J3 = N3(I)                                                        STRAIN     83
      DA11 = DA1(J1) + DN1(I)*(DA1(J2)-DA1(J1))                         STRAIN     84
      DB11 = DB1(J1) + DN1(I)*(DB1(J2)-DB1(J1))                         STRAIN     85
      DB22 = DB2(J1) + DN1(I)*(DB2(J2)-DB2(J1))                         STRAIN     86
      DA22 = DA2(J3) + DN3(I)*(DA2(J3+1)-DA2(J3))                       STRAIN     87
      ZR=ZI                                                             STRAIN     88
      IF(NETAG(I) .EQ. 1)ZR=ZL                                          STRAIN     89
      EPSM1 = (DA11 -      ZR*DB11)*GI11(I)                             STRAIN     90
      EPSM2 = (DA22 -      ZR*DB22)*GI22(I)                             STRAIN     91
   45 EPSS1(I)=SQRT(1.0+2.0*EPSM1)-1.0                                  STRAIN     92
      EPSS2(I)=SQRT(1.0+2.0*EPSM2)-1.0                                  STRAIN     93
      EPSANG(I)=SQRT(1.0+ASA(I)*EPSM1+ASA(I)*EPSM2)-1.0                 STRAIN     94
      EPSANR(I)=SQRT(1.0+BSB(I)*EPSM1+ASB(I)*EPSM2)-1.0                 STRAIN     95
   46 CONTINUE                                                          STRAIN     96
C         COMPONENTS OF VECTOR DISPLACEMENT                             STRAIN     97
      D1=D1+(GR2*DR(N01)+GR1*DR(N02))                                   STRAIN     98
      D2=D2+(GN2*DZ(NG1)+GN1*DZ(NG2))                                   STRAIN     99
      IF(LINK .EQ. 1)GOTO 71                                            STRAIN    100
   50 WRITE(6,60) NCYCLE,TIME                                           STRAIN    101
```

```
      DO 70 I=1,NSTRN                                                    STRAIN   102
      ALEN=SHIMEN                                                        STRAIN   103
      IF(METAG(I).EQ.1)GOTO 67                                           STRAIN   104
      ALEN=SHOLTED                                                       STRAIN   105
   67 WRITE(6,65)ETAG2(I),PN(I),ALEN,EPSS1(I),EPSS2(I),                  STRAIN   106
     1ANGLE(I),EPSANG(I),ANGLE(I),EPSANG(I)                              STRAIN   107
   70 CONTINUE                                                           STRAIN   108
C        PRINTS OF EXTREME STRAINS AND MAXIMUM DEFLECTIONS               STRAIN   109
      WRITE(6,930)TIME,NCYCLE                                            STRAIN   110
      DO 921 L=1,8                                                       STRAIN   111
  921 WRITE(6,932)LEPSX(L),EPSXT(L),NEPSXT(L)                            STRAIN   112
      WRITE(6,931)NCONT,NCYCLE                                           STRAIN   113
      DO 922 L=1,8                                                       STRAIN   114
  922 WRITE(6,933)LEPSX(L),EPSX(L),TEPSX(L),NEPSX(L)                     STRAIN   115
      WRITE(6,941)TIME,NCYCLE,DEFXT,NDEFXT                               STRAIN   116
      WRITE(6,942)NCONT,NCYCLE,UEFX,NDEFX,TDEFX                          STRAIN   117
      GOTO 71                                                            STRAIN   118
  930 FORMAT(//20X,'EXTREME STRAINS AT TIME=',1PE13.6,                   STRAIN   119
     1 '.  TIME CYCLE=',I5,/41X,'STRAIN',7X,'N')                         STRAIN   120
  931 FORMAT(20X,'EXTREME STRAINS BETWEEN CYCLES',I5,' AND ',I5,/        STRAIN   121
     1 32X,'STRAIN',9X,'TIME',9X,'N')                                    STRAIN   122
  932 FORMAT(25X,A10,F13.8,2X,I5)                                        STRAIN   123
  933 FORMAT(16X,A10,F13.8,1PE15.6,2X,I5)                                STRAIN   124
  941 FORMAT(/10X,'MAXIMUM DEFLECTION AT TIME=',1PE13.6,'.  CYCLE=',     STRAIN   125
     1 I5,/13X,'DEFLECTION=',E15.6,5X,'(N = ',I2,')')                    STRAIN   126
  942 FORMAT(10X,'MAXIMUM DEFLECTION BETWEEN CYCLES',I5,' AND',I5/,13X,  STRAIN   127
     1 'DEFLECTION=',1PE15.6,5X,'(N = ',I2,')',10X,      'TIME=',E15.6)  STRAIN   128
   71 RETURN                                                             STRAIN   129
C                                                                        STRAIN   130
   60 FORMAT(//10H TIME STEP,I5,3X,5HTIME=,E16.8//14X,15HSURFACE STRAINS STRAIN   131
     1 ,37X,19HSTRAIN GAGE READING//10X,4HETA2,14X,1HN,5X,               STRAIN   132
     2HFACE,8X,7HANGLE 0,10X,8HANGLE 90,6X,5HANGLE,16X,5HANGLE/)         STRAIN   133
   65 FORMAT(F15.3, F15.3, 4X ,A5,1X,2(2X,E15.8),2(2X,F6.2,2X,E15.8))    STRAIN   134
      END                                                                STRAIN   135
      SUBROUTINE ROUNDR                                                  ROUNDR     2
C                  *CALL MAIN HERE                                       TAP        1
C        IHCE1=1,2,CR3.  IHCE2=1,2,3,OR4.  8/28/74  1/7/76               ROUNDR     4
C        INSERT ONE EXTERNAL R AND Z, BOTH ENDS.  8/23/74                ROUNDR     5
      DELTR=R(N1R+1) - R(N1R)                                            ROUNDR     6
      DELTZ=Z(N1R+1) - Z(N1R)                                            ROUNDR     7
      IF(IHCE1 .EQ. 1) GOTO 11                                           ROUNDR     8
      IF(IHCE1 .EQ. 3) GOTO 13                                           ROUNDR     9
      IF(IHCE1 .NE. 2)GOTO5                                              ROUNDR    10
      Z(N1R+1)=Z(N1R+1)                                                  ROUNDR    11
      Z(N1R-1)=Z(N1R)-DELTZ                                              ROUNDR    12
      GOTO 20                                                            ROUNDR    13
    5 WRITE(6,999)IHCE1,IHCE2                                            ROUNDR    14
  999 FORMAT(' ERROR IN ROUNDR.  IHCE1,IHCE2 =',2I5)                     ROUNDR    15
      STOP' ROUNDR'                                                      ROUNDR    16
   11 TEMP=2.0*(DELTZ*SNR(N1R)-DELTR*SNR(N1R))                           ROUNDR    17
      R(N1R-1) = R(N1R+1) + SNR(N1R)*TEMP                                ROUNDR    18
      Z(N1R-1) = Z(N1R+1) - SNR(N1R)*TEMP                                ROUNDR    19
      GOTO 20                                                            ROUNDR    20
   13 R(N1R-1) = R(N1R) - DELTR                                          ROUNDR    21
      Z(N1R-1) = Z(N1R) - DELTZ                                          ROUNDR    22
   20 DELTR = R(N2R-1) - R(N2R)                                          ROUNDR    23
      DELTZ = Z(N2R-1) - Z(N2R)                                          ROUNDR    24
      IF(IHCE2 .EQ. 1) GOTO 21                                           ROUNDR    25
      IF(IHCE2 .EQ. 2) GOTO 22                                           ROUNDR    26
      IF(IHCE2 .EQ. 3) GOTO 23                                           ROUNDR    27
      IF(IHCE2 .EQ. 4)GOTO 30                                            ROUNDR    28
      GOTO 6                                                             ROUNDR    29
   21 TEMP=2.0*(DELTZ*SNR(N2R)-DELTR*SNR(N2R))                           ROUNDR    30
```

```
      R(N2R+1) = R(N2R-1) + SNR(N2R)*TEMPU                              RCLNDU  31
      Z(N2R+1) = Z(N2R-1) - SNR(N2R)*TEMPU                              RCLNDU  32
      GOTO 30                                                           RCLNDU  33
   22 R(N2R+1) = R(N2R-1)                                               RCLNDU  34
      Z(N2R+1) = Z(N2R)-DELTZ                                           RCLNDU  35
      GOTO 30                                                           RCLNDU  36
   23 R(N2R+1) = R(N2R) - DELTR                                         RCLNDU  37
      Z(N2R+1) = Z(N2R) - DELTZ                                         RCLNDU  38
   30 RETURN                                                            RCLNDU  39
      END                                                               RCLNDU  40
      SUBROUTINE RCLNDU                                                 RCLNDU   2
C                    *CALL MAIN HERE                                    TAB      1
C        IRCE1=1,2,OR3.  IRCE2=1,2,3,OR4.  R/2R/74  1/7/76              RCLNDU   4
C        INSERT DR AND DZ AT ENDS.  ONE EXTERNAL POINT.  R/23/74        RCLNDU   5
      IF(IRCE1 .EQ. 1) GOTO 11                                          RCLNDU   6
      IF(IRCE1 .EQ. 3) GOTO 13                                          RCLNDU   7
      IF(IRCE1 .NE. 2)GOTO 5                                            RCLNDU   8
      DR(N1R-1)=DR(N1R+1)                                               RCLNDU   9
      DZ(N1R-1)=-DZ(N1R+1)                                              RCLNDU  10
      DZ(N1R)=0.0                                                       RCLNDU  11
      GOTO 20                                                           RCLNDU  12
    5 WRITE(6,999)IRCE1,IRCE2                                           RCLNDU  13
  999 FORMAT(' ERROR IN RCLNDU.  IRCE1,IRCE2 =',2I5)                    RCLNDU  14
      STOP' RCLNDU'                                                     RCLNDU  15
   11 TEMPU=2.0*(SNR(N1R)*DZ(N1R+1)-SNK(N1R)*DR(N1R+1))                 RCLNDU  16
      DR(N1R-1) = DR(N1R+1) + SNR(N1R)*TEMPU                            RCLNDU  17
      DZ(N1R-1) = DZ(N1R+1) - SNR(N1R)*TEMPU                            RCLNDU  18
      DR(N1R)=0.0                                                       RCLNDU  19
      DZ(N1R)=0.0                                                       RCLNDU  20
      GOTO 20                                                           RCLNDU  21
   13 DR(N1R-1) = -DR(N1R+1)                                            RCLNDU  22
      DZ(N1R-1) = -DZ(N1R+1)                                            RCLNDU  23
      DR(N1R)=0.0                                                       RCLNDU  24
      DZ(N1R)=0.0                                                       RCLNDU  25
   20 IF(IRCE2 .EQ. 1) GOTO 21                                          RCLNDU  26
      IF(IRCE2 .EQ. 2) GOTO 22                                          RCLNDU  27
      IF(IRCE2 .EQ. 3) GOTO 23                                          RCLNDU  28
      IF(IRCE2 .EQ. 4)GOTO 30                                           RCLNDU  29
      GOTO 5                                                            RCLNDU  30
   21 TEMPU = 2.0*(SNR(N2R)*DZ(N2R-1) - SNR(N2R)*DR(N2R-1))             RCLNDU  31
      DR(N2R+1) = DR(N2R-1) + SNR(N2R)*TEMPU                            RCLNDU  32
      DZ(N2R+1) = DZ(N2R-1) - SNR(N2R)*TEMPU                            RCLNDU  33
      DR(N2R)= 0.0                                                      RCLNDU  34
      DZ(N2R)= 0.0                                                      RCLNDU  35
      GOTO 30                                                           RCLNDU  36
   22 DR(N2R+1) = DR(N2R-1)                                             RCLNDU  37
      DZ(N2R+1) = -DZ(N2R-1)                                            RCLNDU  38
      DZ(N2R)=0.0                                                       RCLNDU  39
      GOTO 30                                                           RCLNDU  40
   23 DR(N2R+1) = -DR(N2R-1)                                            RCLNDU  41
      DZ(N2R+1) = -DZ(N2R-1)                                            RCLNDU  42
      DR(N2R) = 0.0                                                     RCLNDU  43
      DZ(N2R) = 0.0                                                     RCLNDU  44
   30 RETURN                                                            RCLNDU  45
      END                                                               RCLNDU  46
      SUBROUTINE SYMTRY                                                 SYMTRY   2
C                    *CALL MAIN HERE                                    TAB      1
      IF(IRCE2 .NE. 2) GOTO 50                                          SYMTRY   4
      RM22R(N2R+1)=RM22R(N2R-1)                                         SYMTRY   5
      RM22K(N2R+1)=-RM22K(N2R-1)                                        SYMTRY   6
      RN1TR(N2R) = RN1TR(N2R-1)                                         SYMTRY   7
      [illegible]                                                       [illegible]
      RN2R(N2R) = RN2K(N2R-1)                                           SYMTRY   9
```

```
   50 IF(IRCF1 .NE. 2)GOTO 60                                          SYMTRY    10
      FM22R(N1R+1)= FM22R(N1R+1)                                       SYMTRY    11
      FM22K(N1R+1)=-FM22K(N1R+1)                                       SYMTRY    12
      FN1TR(N1R+1)= FN1TR(N1R)                                         SYMTRY    13
       FN2R(N1R+1)= -FN2R(N1R)                                         SYMTRY    14
       FN2K(N1R+1)=  FN2K(N1R)                                         SYMTRY    15
   60 RETURN                                                           SYMTRY    16
      END                                                              SYMTRY    17
      SUBROUTINE ARINIT(N)                                             ARINIT    2
      STOP *ARINIT DELETED*                                            ARINIT    3
      END                                                              ARINIT    4
      SUBROUTINE KINET                                                 KINET     2
C                  *CALL MAIN HERE                                     TAP       1
      CINET=0.                                                         KINET     4
      DO 83 N=N1V,N2V                                                  KINET     5
      CN=1.0                                                           KINET     6
      IF(N .EQ. N1R)CN=0.5                                             KINET     7
      IF(N.EQ.N2R)CN=0.5                                               KINET     8
      CINET=CINET+(DR(N)**2+DZ(N)**2)/TEMP(N) *CN                      KINET     9
   83 CONTINUE                                                         KINET     10
      IF(IRCF1 .EQ. 2)CINET=2.0*CINET                                  KINET     11
      IF(IRCF2 .NE. 2)GOTO 86                                          KINET     12
      CINET=2.0*CINET                                                  KINET     13
   86 CINER=CINES                                                      KINET     14
      CINES=0.5*CA*CINET                                               KINET     15
      CINET=0.5*(CINES+CINER)                                          KINET     16
      RETURN                                                           KINET     17
      END                                                              KINET     18
      SUBROUTINE PBCHK                                                 PBCHK     2
C                  *CALL MAIN HERE                                     TAP       1
      DO 20 N=N1V,N2V                                                  PBCHK     4
      CN=0.50                                                          PBCHK     5
      IF(N .EQ. N1R)CN=0.25                                            PBCHK     6
      IF(N.EQ.N2R)CN=0.25                                              PBCHK     7
      DR=DR(N)*SNR(N)+DZ(N)*SNK(N)                                     PBCHK     8
      ENS=ENS-CN*DN*P(N)                                               PBCHK     9
   20 CONTINUE                                                         PBCHK     10
      RETURN                                                           PBCHK     11
      END                                                              PBCHK     12
      SUBROUTINE DAMP                                                  DAMP      2
C                  *CALL MAIN HERE                                     TAP       1
C                                                                      DAMP      4
C         CHECK FOR START OF DAMPING                                   DAMP      5
      IF(NCYCLE .LT. NDAMP)RETURN                                      DAMP      6
      IF(NCYCLE .GT. NDAMP)GOTO 16                                     DAMP      7
C         MONITOR THE DAMPING                                          DAMP      8
      WRITE(6,101)NCYCLE,TIME                                          DAMP      9
  101 FORMAT(/' INITIATED DAMPING.  NCYCLE = ',I6,10X,'TIME =',1PE15.7) DAMP      10
      LOAD = 0                                                         DAMP      11
      IPRESS = 0                                                       DAMP      12
      DO 15 N=1,NN                                                     DAMP      13
      P(N)=0.0                                                         DAMP      14
   15 CONTINUE                                                         DAMP      15
   16 CONTINUE                                                         DAMP      16
      IF(CINES-CINER)20,20,40                                          DAMP      17
   20 TDAMP=TDAMP+CINET                                                DAMP      18
      WRITE(6,102)NCYCLE,CINES,CINER,CINET,TDAMP,CINER                 DAMP      19
  102 FORMAT(/' STOPPING PROCEDURE',NCYCLE,CINES,CINER,CINET,TDAMP,CINER/ DAMP      20
     1 / 1I6,1P5E15.7)                                                 DAMP      21
      IF(CINET+CINER .LE. REACT*TDAMP)GOTO 50                          DAMP      22
      DO 30 N=1,NN                                                     DAMP      23
      DR(N)=0.0                                                        DAMP      24
      DZ(N)=0.0                                                        DAMP      25
```

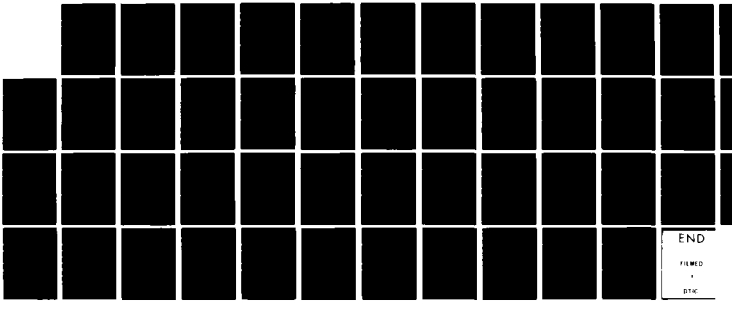
```
   30 CONTINUE                                                          DAMP      26
      CINEP=CINFT                                                       DAMP      27
      CINES1 = CINEP                                                    DAMP      28
      CINES2 = CINES                                                    DAMP      29
      CINES=0.0                                                         DAMP      30
      NCYCLE = NCYCLE + 1                                               DAMP      31
      TIME = TIME + DELTAT                                              DAMP      32
      CALL MOTION                                                       DAMP      33
      IF(CINES .LE. CINES1)GOTO 39                                      DAMP      34
      CALLDESTEP                                                        DAMP      35
   39 CALL PDATA(2)                                                     DAMP      36
   40 TDAMP=TDAMP+C2*CINES                                              DAMP      37
   45 RETURN                                                            DAMP      38
   50 WRITE(6,100) NCYCLE                                               DAMP      39
      MAXC=NCYCLE                                                       DAMP      40
      NC3DP(NN3D)=NCYCLE                                                DAMP      41
      CALL PDATA (2)                                                    DAMP      42
      GOTO 45                                                           DAMP      43
  100 FORMAT(1H1,10X,30HRUN SELF-TERMINATED TIME STEP,I5)               DAMP      44
      END                                                               DAMP      45
      SUBROUTINE DESTEP                                                 DESTEP     2
C                 *CALL MAIN HERE                                       TAP       1
C        DECREASE DELTAT  (IT IS TOO LARGE FOR STOPPING PROCEDURE)      DESTEP     4
      DELTAT=SQRT(CINES1/CINES)*DELTAT                                  DESTEP     5
      DSGOLD=DELSQ                                                      DESTEP     6
      C1OLD=C1                                                          DESTEP     7
      DELSQ=DELTAT**2                                                   DESTEP     8
      C2=2.0*DELTAT*DAMPF/GAM2                                          DESTEP     9
      C1=C2/(4.0+C2)                                                    DESTEP    10
      DELG=DELSQ/DSGOLD                                                 DESTEP    11
      DELS=DELP*(1.0-C1)/(1.0-C1OLD)                                    DESTEP    12
      CINES=CINES*DELS**2/DELP                                          DESTEP    13
      CINFT=0.5*(CINES+CINEP)                                           DESTEP    14
      PLAST=TNRG-CINFT-STREN-TDAMP                                      DESTEP    15
      DO 10 N=1,NN                                                      DESTEP    16
      TEMP(N)=DELP*TEMP(N)                                              DESTEP    17
      DR(N)=DELS*DR(N)                                                  DESTEP    18
      DZ(N)=DELS*DZ(N)                                                  DESTEP    19
   10 CONTINUE                                                          DESTEP    20
      RETURN                                                            DESTEP    21
      END                                                               DESTEP    22
      SUBROUTINE PDATA(LINK)                                            PDATA      2
C                 *CALL MAIN HERE                                       TAP       1
      DIMENSION DAT(20)                                                 PDATA      4
C       PDATA SELECTS AND WRITES DATA ON TAPE(NPLOT) FOR THE REPSIL     PDATA      5
C       PLOTTING PROGRAM                                                PDATA      6
C                                                                       PDATA      7
      GOTO (10,40,50,60),LINK                                           PDATA      8
   10 NN3D=1                                                            PDATA      9
      II = 2*NSTRN + 8                                                  PDATA     10
C        INITIATE PLOT FILE.  RESTART CAPABILITIES DELETED.             PDATA     11
   15 WRITE(NPLOT)ETAG2,GN,NSTRN                                        PDATA     12
      WRITE(NPLOT)(ETAG2(I),FR(I),NETAG(I),I=1,NSTRN)                   PDATA     13
      NPTS=N2H-N1P+1                                                    PDATA     14
      WRITE(NPLOT)NCYCLE,TIME,NPTS,(R(N),Z(N),N=N1P,N2H)                PDATA     15
C                                                                       PDATA     16
      DO 25 I=1,II                                                      PDATA     17
      DAT(I)=0.0                                                        PDATA     18
   25 CONTINUE                                                          PDATA     19
      IF(LOAD) 30,30,35                                                 PDATA     20
   30 DAT(5)=TNRG                                                       PDATA     21
      DAT(6)=PLAS                                                       PDATA     22
      DAT(7)=TNRG                                                       PDATA     23
```

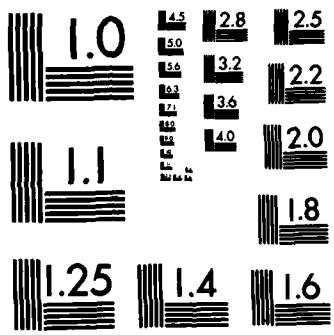
```
      DAT(8)=TMAG                                                       PDATA   24
   35 IFLAG=1                                                           PDATA   25
      WRITE(NPLOT) IFLAG                                                PDATA   26
      WRITE(NPLOT) NCYCLE,(DAT(I),I=1,11)                               PDATA   27
      GOTO 100                                                          PDATA   28
C                                                                       PDATA   29
   40 DAT(1)=TIME                                                       PDATA   30
      DAT(2)=E1                                                         PDATA   31
      DAT(3)=E2                                                         PDATA   32
      DAT(5)=CINET                                                      PDATA   33
      DAT(6)=STPEN+CINET                                                PDATA   34
      DAT(7)=TMAG                                                       PDATA   35
      DAT(8)=DAT(6)+TDAMP                                               PDATA   36
      J = 9                                                             PDATA   37
      DO 45 I=1,NSTRN                                                   PDATA   38
      DAT(J)=EPSS1(I)                                                   PDATA   39
      DAT(J+1)=EPSS2(I)                                                 PDATA   40
      J=J+2                                                             PDATA   41
   45 CONTINUE                                                          PDATA   42
      IFLAG=1                                                           PDATA   43
      WRITE(NPLOT) IFLAG                                                PDATA   44
      WRITE(NPLOT) NCYCLE,(DAT(I),I=1,11)                               PDATA   45
C          CHECK FOR 3D PLOT                                            PDATA   46
      IF(NCYCLE .NE. NC3D(NN3D))GOTO 100                                PDATA   47
      NN3D=NN3D+1                                                       PDATA   48
      IFLAG=2                                                           PDATA   49
      WRITE(NPLOT) IFLAG                                                PDATA   50
      NPTS=N2B-N1B+1                                                    PDATA   51
      WRITE(NPLOT)NCYCLE,TIME,NPTS,(R(N),Z(N),N=N1B,N2B)                PDATA   52
      GOTO 100                                                          PDATA   53
C                                                                       PDATA   54
C   50 END FILE NPLOT & DELETED, MARKED RESTART POSITIONS FOR BRLESC    PDATA   55
   50 CONTINUE                                                          PDATA   56
      GOTO 100                                                          PDATA   57
C                                                                       PDATA   58
   60 IFLAG=99999                                                       PDATA   59
      WRITE(NPLOT) IFLAG                                                PDATA   60
  100 RETURN                                                            PDATA   61
      END                                                               PDATA   62
      SUBROUTINE PRESS                                                  PRESS    2
C                  *CALL MAIN HERE                                      TAP      1
C          CONSTANT PRESSURE = P0.                                      PRESS    4
      DATA IPRESS/0/                                                    PRESS    5
      IF(IPRESS .GT. 0)GOTO 2                                           PRESS    6
      IPRESS = 1                                                        PRESS    7
      READ(5,100)P0                                                     PRESS    8
      WRITE(6,105)P0                                                    PRESS    9
    2 DO 5 N=N1V,N2V                                                    PRESS   10
    5 P(N)=P0                                                           PRESS   11
      RETURN                                                            PRESS   12
  100 FORMAT(F12.6)                                                     PRESS   13
  105 FORMAT(//36X,'CONSTANT PRESSURE LOADING.  P0=',1PE15.6/)          PRESS   14
      END                                                               PRESS   15
      SUBROUTINE INGEOM                                                 INGEOM   2
C        SIGNALS TO PROGRAM FROM INGEOM.                                INGEOM   3
C          BEAM  (IB>0).                                                INGEOM   4
C           SLAB SYMMETRY  (RADIUS = 0.0   IB=0).                       INGEOM   5
C          AXIAL SYMMETRY  (RADIUS > 0.0,  IB=0).                       INGEOM   6
C        INGEOM FOR STRAIGHT BEAM OR SLAB SYMMETRIC FLAT SLAB           INGEOM   7
C     EVALUATE THE INITIAL GEOMETRY                                     INGEOM   8
C                  *CALL MAIN HERE                                      TAP      1
C        SET PROGRAM TO USE GAUSSIAN INTEGRATION IN INGEOM.             INGEOM  10
      IGAUSS = 1                                                        INGEOM  11
```

```
C                                                                       INGEOM   12
      RADIUS = 0.0                                                      INGEOM   13
C RADIUS =0.0 IS A SIGNAL TO USE SLAB SYMMETRY OR BEAM                  INGEOM   14
C                                                                       INGEOM   15
      READ(5,100) SLAHL, SLAHW, IB                                      INGEOM   16
C                                                                       INGEOM   17
      DETA1 = 2.0*SLAHW                                                 INGEOM   18
C ENERGY IS COMPUTED FOR A WIDTH OF DETA1.                              INGEOM   19
C DETA1 TENTATIVELY SET AT 2*SLAHW.  IF IB > 0 DETA1 SET TO 1.0 IN START INGEOM  20
C                                                                       INGEOM   21
      DETA2 = SLAHL/FLOAT(NMESH)                                        INGEOM   22
C                                                                       INGEOM   23
C        EVALUATE(R(N),Z(N),N=N1B,N2B)                                  INGEOM   24
C                                                                       INGEOM   25
      DO 10 N=N1B,N2B                                                   INGEOM   26
      R(N) = 0.0                                                        INGEOM   27
      Z(N)=FLOAT(N-N1B)*DETA2                                           INGEOM   28
   10 CONTINUE                                                          INGEOM   29
C                                                                       INGEOM   30
      IF(IB .LE. 0) RETURN                                              INGEOM   31
      READ(5,98)ZU                                                      INGEOM   32
      WRITE(6,97) SLAHL,ZU                                              INGEOM   33
      DO 12 K=1,LAYER                                                   INGEOM   34
      READ(5,98) WIDTHK,DZETAK,ZETA(K)                                  INGEOM   35
      W(K)=WIDTHK*DZETAK                                                INGEOM   36
      WRITE(6,99) K,WIDTHK,DZETAK,ZETA(K),W(K)                          INGEOM   37
   12 CONTINUE                                                          INGEOM   38
   97 FORMAT(/' INGEOM FOR BEAM OF LENGTH',E15.7/                       INGEOM   39
     * ' DISTANCE OF CENTROID FROM TOP OF BEAM IS',E15.7/               INGEOM   40
     1 23X,'DIMENSIONS OF LAYERS'/                                      INGEOM   41
     2 2X,'K',7X,'WIDTHK', 9X,'DELHEIGHT',6X,'ZETA(K)',8X,'AREA'/)      INGEOM   42
   98 FORMAT(3E10.2)                                                    INGEOM   43
   99 FORMAT(I3,2X,4E15.7)                                              INGEOM   44
      RETURN                                                            INGEOM   45
  100 FORMAT(2E10.4,I5)                                                 INGEOM   46
      END                                                               INGEOM   47
      SUBROUTINE INNORM                                                 INNORM    2
C                  *CALL MAIN HERE                                      TAPE     1
C              NEW CONDITIONS ON FIXED ENDS 12/12/73                    INNORM    4
C              INNORM COMPUTES THE INITIAL NORMAL AT CLAMPED ENDS       INNORM    5
C         THIS VERSION ASSUMES A QUADRATIC THRU THE THREE END POINTS    INNORM    6
      IF(IBCE1 .NE. 1) GOTO 5                                           INNORM    7
C         END1 N=N1B, IS CLAMPED                                        INNORM    8
      R2=R(N1B+2)-4.0*R(N1B+1)+3.0*R(N1B)                               INNORM    9
      Z2=Z(N1B+2)-4.0*Z(N1B+1)+3.0*Z(N1B)                               INNORM   10
      D=SQRT(R2**2+Z2**2)$ SNR(N1B)=Z2/D$ SNR(N1B)=-R2/D                INNORM   11
    5 IF(IBCE2 .NE. 1)GOTO 6                                            INNORM   12
C         END2 N=N2B, IS CLAMPED                                        INNORM   13
      R2=R(N2B-2)-4.0*R(N2B-1)+3.0*R(N2B)                               INNORM   14
      Z2=Z(N2B-2)-4.0*Z(N2B-1)+3.0*Z(N2B)                               INNORM   15
      D=SQRT(R2**2+Z2**2)$ SNR(N2B)=Z2/D$ SNR(N2B)=-R2/D                INNORM   16
    6 RETURN                                                            INNORM   17
      END                                                               INNORM   18
```

AD-A122 336 RPSL1D (A ONE-DIMENSIONAL VERSION OF REPSIL)(U) ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERDEEN PROVI.. J D WORTMAN NOV 82 ARBRL-MR-03221 2/2

UNCLASSIFIED SBI-AD-F300 132 F/G 19/1 NL





APPENDIX D

PLOTTING PROGRAM FOR RPSL1D

The plotting program is an independent program. The RPSL1D program stores data for plotting through unit 3 which is equivalenced to file BLOT on the PROGRAM card. The plotting program, which we have catalogued in file RPSL1DPLOT, reads data from unit 3 also equivalenced to file BLOT. For some runs it may be advantageous to catalogue BLOT and repeat the plotting program to reproduce plots that were unsatisfactory because of scaling (input card) or system problems.

This program uses the BRL plotting subroutines described in ARDC TR6 (BRLESC FORTRAN PLOTTING SUBROUTINES, Monte W. Coleman, John V. Lanahan, July 1970) as amended for CDC by local publication SPB-6-78, May 2, 1978. Conversion to SCOOP, the plotting system used in Reference 2, or another plotting system, would not be difficult.

The main program, RP1PLT, reads binary data from unit 3 (equated to BLOT) and controls the program flow. If the input variable IFLAG is 1, an array of data for one time point is read and stored. If IFLAG equals 2, different data is read and PLOT3D is called. If IFLAG equals 99999, or certain abnormal conditions occur, the program calls subroutine GRAPH.

Subroutine PLOT3D reads a control card on the initial entry containing DEFLM, SOFC, and SF with FORMAT (3E10.3), and sets up scaling for isometric plots of the "center line deflection profile" of Reference 2. This is a plot of the initial position of the reference curve, and the current position of the curve with the difference magnified by DEFLM. If SF is greater than zero, it is the reciprocal of the scale factor between the internal length measure and inches on the plotting surface. If the input SF is not greater than zero, the program assigns an SF that attempts to scale the plot of the initial curve into an SOFC by SOFC inch square which also includes the origin.

Subroutine GRAPH plots displacement increments DR and DZ vs. time at point (ETAD1, ETAD2) and the energy balance plot: time vs. kinetic energy, kinetic energy plus strain energy, total energy, and total energy less damping work. Subroutine GRAPH calls subroutine STRAIN to produce NSTRN plots of strains in the coordinate directions vs. time at points prescribed on input cards 13. If the PLOTP option is included, GRAPH also produces the NNPE prescribed plots of P(N) vs. time.

All the COMMON variables are included in COMDECK MAIN. The longer arrays are put in LEVEL 2 (a special version of FIXSCA was inserted to use them); if more than 3000 time cycles are recorded, these arrays and MAXC must be increased to plot them.

The following list is the COMPILE file image of RPSL1DPLOT formed through UPDATE. This listing gives the correct UPDATE card identifiers.

```
      PROGRAM RP1PLT(INPLT,OUTPUT,PLOT,TAPE13,TAPE5=INPUT,TAPE6=OUTPUT,     RP1PLT     2
     1  TAPE3=PLOT)                                                         RP1PLT     3
C        PLOTTING FOR 1-C PEPSIL.                                          RP1PLT     4
C        PEPSID PLOTTING PACKAGE  (CALCOMP PLOTTING,  SPR=6-78)            RP1PLT     5
C        IF NCYCL>MAXC, INCREASE MAXC AND ARRAYS IN PARRAY AND PLOTP.      RP1PLT     6
C          PLOTTING P(N).  9/30/75  (PD(3002,9))                           RP1PLT     7
C                                                                          RP1PLT     8
      COMMON ETAD2(6),PN(6),NETAG(6),NSTRN,MAXC                            MAIN       2
      COMMON RR(103),ZZ(103),Y(103),Z(103)                                 MAIN       3
      COMMON DAT(20),NCYCLE,TIME,              ETAD2,   GN,NCYCL           MAIN       4
      COMMON NNPE,NPE(9),PDAT(9)                                           MAIN       5
      COMMON /PARRAY/                                                      MAIN       6
     1  TIM( 3002),    U2( 3002),   U3( 3002),  CIN( 3002),  STC( 3002),  MAIN       7
     2  TNR( 3002),DAMPLT( 3002),EPSS1(18012),EPSS2(18012)                 MAIN       8
      COMMON/PLOTP/PD( 3002,9)                                             MAIN       9
      LEVEL2,TIM,PD                                                        MAIN      10
      MAXC = 3002                                                          RP1PLT    10
       NNPE = 0                                                            RP1PLT    11
      NCYCL=0                                                              RP1PLT    12
      NPLOT=3                                                              RP1PLT    13
      REWIND NPLOT                                                         RP1PLT    14
C                                                                          RP1PLT    15
      READ(NPLOT) ETAD2,GN,NSTRN                                           RP1PLT    16
      READ(NPLOT) (ETAG2(I),PN(I),NETAG(I),I=1,NSTRN)                      RP1PLT    17
C     READ(NPLOT)NNPE,(NPE(I),I=1,NNPE)      ACTIVATE FOR PLOTP            RP1PLT    18
      READ(NPLOT) NCYCLE,TIME,N1,(RR(N),ZZ(N),N=1,N1)                      RP1PLT    19
      II=2*NSTRN+8                                                         RP1PLT    20
C                                                                          RP1PLT    21
C       SAVE INITIAL SHAPE NEEDED FOR DEFLECTION MAGNIFICATION IN PLOT3D   RP1PLT    22
      DO 5 N=1,N1                                                          RP1PLT    23
      Y(N)=ZZ(N)                                                           RP1PLT    24
      Z(N)=RR(N)                                                           RP1PLT    25
    5 CONTINUE                                                             RP1PLT    26
      CALL PLOT3D(N1)                                                      RP1PLT    27
C                                                                          RP1PLT    28
   10 READ(NPLOT) IFLAG                                                    RP1PLT    29
      IF(EOF(NPLOT) .NE. 0) GOTO 28                                        RP1PLT    30
      IF(IFLAG .EQ. 99999)GOTO 30                                          RP1PLT    31
      IF(IFLAG .EQ. 1)GOT  20                                              RP1PLT    32
      IF(IFLAG .EQ. 2)GOTO 25                                              RP1PLT    33
      WRITE(6,911)IFLAG,NCYCL,NCYCLE                                       RP1PLT    34
  911 FORMAT(//' BAD SIGNAL FROM TAPE.  IFLAG,NCYCL,NCYCLE=',3I10)         RP1PLT    35
C        IF RUN FAILS, GET PARTIAL PLOT.                                   RP1PLT    36
      GOTO 28                                                              RP1PLT    37
C                                                                          RP1PLT    38
   20 READ (NPLOT) NCYCLE,(DAT(I),I=1,II)                                  RP1PLT    39
C    1 ,(PDAT(J),J=1,NNPE)                    ACTIVATE FOR PLOTP          RP1PLT    40
      IF(EOF(NPLOT) .NE. 0) GOTO 28                                        RP1PLT    41
      NCYCL=NCYCL+1                                                        RP1PLT    42
      IF(NCYCL .GT. MAXC)GOTO 28                                           RP1PLT    43
       TIM(NCYCL)=DAT(1)                                                   RP1PLT    44
        U3(NCYCL)=DAT(2)                                                   RP1PLT    45
        U2(NCYCL)=DAT(3)                                                   RP1PLT    46
      CIN (NCYCL)=DAT(5)                                                   RP1PLT    47
      STC (NCYCL)=DAT(6)                                                   RP1PLT    48
      TNR (NCYCL)=DAT(7)                                                   RP1PLT    49
      DAMPLT(NCYCL)=DAT(8)                                                 RP1PLT    50
      DO 22 I=9,II,2                                                       RP1PLT    51
      J=NCYCL + MAXC*(I-9)/2                                               RP1PLT    52
      EPSS1(J)=DAT(I)                                                      RP1PLT    53
      EPSS2(J)=DAT(I+1)                                                    RP1PLT    54
   22 CONTINUE                                                             RP1PLT    55
```

```
      IF(NNPE .EQ. 0)GOTO 10
      DO 23 I=1,NNPE
   23 PD(NCYCL,I) = PDAT(I)
      GOTO 10
   25 READ(NPLOT) NCYCLE,TIME,N1,(RR(N),ZZ(N),N=1,N1)
      IF(EOF(NPLOT) .NE. 0) GOTO 28
      CALL PLOT3D(N1)
      GOTO 10
C
   28 WRITE(6,100) MAXC
  100 FORMAT(//20H ERROR NCYCL > MAXC=,I5/)
C        THE LAST SET OF DATA IS NO GOOD.  PLOT THE REST.
      NCYCL = NCYCL - 1
C
   30 CALL GRAPH
      CALL PLTPGE
      CALL EXIT
      END
      SUBROUTINE PLOT3D(I2)
      COMMON ETAG2(6),PN(6),NETAG(6),NSTRN,MAXC
      COMMON RR(103),ZZ(103),Y(103),Z(103)
      COMMON DAT(20),NCYCLE,TIME,               ETAD2,   QN,NCYCL
      COMMON NNPE,NPE(9),PDAT(9)
      COMMON /MARRAY/
     1  TIM( 3002),    U2( 3002),    U3( 3002),  CIN( 3002),  STC( 3002),
     2  TNR( 3002),DAMPLT( 3002),EPSS1(18012),EPSS2(18012)
      COMMON/PLOTP/PD( 3002,9)
      LEVEL2,TIM,PD
      DIMENSION          X1(103),X2(103),          HEAD2(3),HEAD3(2)
      DIMENSION LABEL(4)
      DATA LABEL/'WORTMAN','B309','X3979','RPIPLT'/
      DATA I/0/
      DATA (HEAD2(IK),IK=1,3)/10HDEFLECTION,10H MAGNIFIED,1H>/
      DATA (HEAD3(IK),IK=1,2)/10HMICROSECON,3HDS>/
      IF(I.EQ.0)GOTO 10
      I=I+1
      YBARN=YBARN+10.0
      IF(I.LE.3)GOTO 30
      CALL PLTPGE
      GOTO 20
   10 XPAGE=12.0
      READ(5,11) DEFLM,SOFC,SF
   11 FORMAT(3F10.3,I5)
C     -------------------- SCALE FACTOR FOR 3D.PLOT ------------------
      YMAX=Y(1)
      YMIN=Y(1)
      ZMAX=Z(1)
      ZMIN=Z(1)
      DO 12 N=1,I2
      YMAX=AMAX1(Y(N),YMAX)
      YMIN=AMIN1(Y(N),YMIN)
      ZMAX=AMAX1(Z(N),ZMAX)
      ZMIN=AMIN1(Z(N),ZMIN)
   12 CONTINUE
      IF(SF .NE. 0.0)GOTO 13
      YS=(YMAX-YMIN)/SOFC
      ZS=(ZMAX-ZMIN)/SOFC
      SF=AMAX1(    YS,ZS)
      IF(SF .GT. 0.0 .AND. SF .LT. 0.99)GOTO 13
      SF=AINT(SF)
      IF(SF .LT. 1.0) SF=1.0
   13 CALL [illegible](XPAGE,29,[illegible],13,LABEL)
      CALL PLTSCA (1.0,1.0,0.0,0.0,1.0,1.0)
```

```
      CALL PLTSYM (0.1,HEAD2(1),0.0,     0.0,-.3)                    PLOT3D     39
      ENCODE(12,15,STEP)DEFLM                                         PLOT3D     40
   15 FORMAT(F10.3,2H >)                                              PLOT3D     41
      CALL PLTSYM (0.1,STEP,0.0,     2.1,-.3)                         PLOT3D     42
      CALL PLTSYM (0.1,9HSCALE 1/>,0.0,     0.0,-.5)                  PLOT3D     43
      ISF=INT(SF)                                                     PLOT3D     44
      ENCODE(4,16,STEP)ISF                                            PLOT3D     45
   16 FORMAT(I2,2H >)                                                 PLOT3D     46
      CALL PLTSYM (0.1,STEP,0.0,     0.9,-.5)                         PLOT3D     47
      SF=1.0/SF                                                       PLOT3D     48
C     * * * * * *   *   *   *   *   *   *   *   *   *   *   *   *   *  PLOT3D     49
   20 YBARN=0.0                                                       PLOT3D     50
      I=1                                                             PLOT3D     51
   30 YBAR=4.0+YBARN                                                  PLOT3D     52
      XBAR=1.0                                                        PLOT3D     53
C     ------------------- CENTER-LINE DEFLECTION PROFILE ------------- PLOT3D     54
      YBAC=YBAR                                                       PLOT3D     55
      CALL PLTSCA(XBAR,YBAC,0.0,0.0,1.0,1.0)                          PLOT3D     56
      X1(1)=0.0                                                       PLOT3D     57
      X2(1)=-SF*Z(1)                                                  PLOT3D     58
      X1(2)=0.0                                                       PLOT3D     59
      X2(2)=SF*Z(1)                                                   PLOT3D     60
      CALL PLTDTS(1.0,X1(1),X2(1),2.0)                                PLOT3D     61
      X2(1)=0.0                                                       PLOT3D     62
      X1(2)=SF*Y(I2)                                                  PLOT3D     63
      X2(2)=0.0                                                       PLOT3D     64
      CALL PLTDTS(1.0,X1(1),X2(1),2.0)                                PLOT3D     65
      K=0                                                             PLOT3D     66
      DO 60 N=1,I2                                                    PLOT3D     67
      K=K+1                                                           PLOT3D     68
      X1(K)=SF*Y(N)                                                   PLOT3D     69
      X2(K)=SF*Z(N)                                                   PLOT3D     70
   60 CONTINUE                                                        PLOT3D     71
      CALL PLTDTS(4.0,X1(1),X2(1),K,0)                                PLOT3D     72
      CALL PLTDTS(3.0,X1(1),X2(1),K,0)                                PLOT3D     73
      K=0                                                             PLOT3D     74
      DO 70 N=1,I2                                                    PLOT3D     75
      K=K+1                                                           PLOT3D     76
      X1(K)=SF*(Y(N)+DEFLM*(ZZ(N)-Y(N)))                              PLOT3D     77
      X2(K)=SF*(Z(N)+DEFLM*(RR(N)-Z(N)))                              PLOT3D     78
   70 CONTINUE                                                        PLOT3D     79
      CALL PLTDTS(2.0,X1(1),X2(1),K,0)                                PLOT3D     80
C     * * * * * *   *   *   *   *   *   *   *   *   *   *   *   *   *  PLOT3D     81
      CALL PLTSYM (.1,6HCYCLE>,0.0,      .2,-3.0)                     PLOT3D     82
      ENCODE (6,101,STEP) NCYCLE                                      PLOT3D     83
  101 FORMAT(I5,1H>)                                                  PLOT3D     84
      WRITE(6,109)NCYCLE                                              PLOT3D     85
  109 FORMAT(' ISOMETRIC PLOT, CYCLE',I6)                             PLOT3D     86
      CALL PLTSYM (.1,STEP,0.0,      .8,-3.0)                         PLOT3D     87
      TI=TIME*1.0E6                                                   PLOT3D     88
      ENCODE (10,105,STEP) TI                                         PLOT3D     89
  105 FORMAT(F8.1,2H >)                                               PLOT3D     90
      CALL PLTSYM (.1,STEP,0.0,      .2,-3.2)                         PLOT3D     91
      CALL PLTSYM (.1,HEAD3(1),0.0,     1.1,-3.2)                     PLOT3D     92
C     * * * * * *   *   *   *   *   *   *   *   *   *   *   *   *   *  PLOT3D     93
      RETURN                                                          PLOT3D     94
      END                                                             PLOT3D     95
      SUBROUTINE GRAPH                                                GRAPH      2
      COMMON ETAG2(6),PN(6),NETAG(6),NSTRN,MAXC                       MAIN       2
      COMMON RR(103),ZZ(103),Y(103),Z(103)                            MAIN       3
      COMMON DAT(20),NCYCLE,TIME,          ETAD2,   GN,NCYCL          MAIN       4
      COMMON ANPE,APE(9),EDAT(9)                                      MAIN       5
      COMMON /PARRAY/                                                 MAIN       6
```

```
     1  TIM( 3002),    U2( 3002),   L3( 3002),  CIN( 3002),  STC( 3002), MAIN        7
     2  TNR( 3002),DAMPLT( 3002),EPSS1(18012),EPSS2(18012)                MAIN        8
      COMMON/PLOTP/PD( 3002,6)                                           MAIN        9
      LEVEL2,TIM,PD                                                      MAIN       10
      DIMENSION          SYM1(4),SYM2(2),         ETA2(3),SY4(3)         GRAPH       4
      DATA(SYM1(I),I=1,4)/10HCOMPONENT ,10HOF VECTOR ,10HDISPLACEME,3HNT GRAPH       5
     1>/,(SYM2(I),I=1,2)/10HTIME (MICR,10HOSECONDS)>/                    GRAPH       6
      DATA(ETA2(I),I=1,3)/10HETA2 =     ,10H       (    ,5H    )>/,     GRAPH       7
     2     (SY4(I),I=1,2)/10H     ENERGY,9H BALANCE>/                   GRAPH       8
C                                                                        GRAPH       9
      CALL PLTPGF                                                        GRAPH      10
      XBAR=3.0                                                           GRAPH      11
      YBAR=2.0                                                           GRAPH      12
      XL=7.9                                                             GRAPH      13
      YL=6.4                                                             GRAPH      14
      N=NCYCL                                                            GRAPH      15
      XS=1.0E6                                                           GRAPH      16
C     ---------------------------- GRAPH ONE ----------------------------- GRAPH      17
      CALL FIXSCA (TIM(1),N,XL,XS1,XMIN1,XMAX1,DX1)                      GRAPH      18
      CALL FIXSCA (U3(1),N,YL,YS1,YMIN1,YMAX1,DY1)                       GRAPH      19
      CALL CONSCA (U2(1),N,YL,YS1,YMIN1,YMAX1,DY1)                       GRAPH      20
      CALL PLTSCA (XBAR,YBAR,XMIN1,YMIN1,XS1,YS1)                        GRAPH      21
      CALL PLTAXS (DX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,4)                    GRAPH      22
      TEM1=XMAX1+DX1/3.0                                                 GRAPH      23
      CALL PLTDT2 (1.0,TIM(1),U2(1),N,0)                                 GRAPH      24
      U2N = U2(N)                                                        GRAPH      25
      CALL PLTSYM (.1,3HDZ>,0.0,     TEM1   ,U2 N )                      GRAPH      26
      CALL PLTDT2 (1.0,TIM(1),U3(1),N,0)                                 GRAPH      27
      U3N = U3(N)                                                        GRAPH      28
      CALL PLTSYM (.1,3HDR>,0.0,     TEM1   ,U3.N )                      GRAPH      29
      CALL LABELA (DX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,XS,1.0)               GRAPH      30
      CALL PLTSCA (XBAR,YBAR,0.0,0.0,1.0,1.0)                            GRAPH      31
      CALL PLTSYM (.1,SYM2(1),0.0,     3.0,-0.6)                         GRAPH      32
      CALL PLTSYM (.1,SYM1(1),90.0,    -1.2,1.4)                         GRAPH      33
      CALL PLTSYM (.1,9HLOCATION>,0.0,     3.0,-1.0)                     GRAPH      34
      CALL PLTSYM (.3,2H0>,0.0,     3.8,-1.1)                            GRAPH      35
      CALL PLTSYM (.3,2H1>,0.0,     6.6,-1.1)                            GRAPH      36
      CALL PLTSYM (.1,ETA2(1),0.0,     4.1,-1.1)                         GRAPH      37
      ENCODE (8,50,STEP)ETAD2                                            GRAPH      38
      CALL PLTSYM (.1,STEP,0.0,     4.8,-1.1)                            GRAPH      39
      ENCODE (8,50,STEP)DA                                               GRAPH      40
      CALL PLTSYM (.1,STEP,0.0,     5.7,-1.1)                            GRAPH      41
C     ---------------------------- GRAPH TWO ----------------------------- GRAPH      42
      YBAR=12.0                                                          GRAPH      43
      CALL PLTSCA(XBAR,YBAR,0.0,0.0,1.0,1.0)                             GRAPH      44
      CALL PLTSYM(.1,SY 4(1),90.0,    -1.0,1.5)                          GRAPH      45
      CALL PLTSYM(.1,SYM2(1),0.0,     3.0,-0.6)                          GRAPH      46
      CALL FIXSCA (CIN(1),N,YL,YS4,YMIN4,YMAX4,DY4)                      GRAPH      47
      CALL CONSCA (STC(1),N,YL,YS4,YMIN4,YMAX4,DY4)                      GRAPH      48
      CALL CONSCA (TNR(1),N,YL,YS4,YMIN4,YMAX4,DY4)                      GRAPH      49
      CALL CONSCA (DAMPLT(1),N,YL,YS4,YMIN4,YMAX4,DY4)                   GRAPH      50
      CALL PLTSCA (XBAR,YBAR,XMIN1,YMIN4,XS1,YS4)                        GRAPH      51
      CALL PLTAXS (DX1,DY4,XMIN1,XMAX1,YMIN4,YMAX4,4)                    GRAPH      52
      CALL LABELA (DX1,DY4,XMIN1,XMAX1,YMIN4,YMAX4,XS,1.0)               GRAPH      53
      CALL PLTDT2(1.0,TIM(1),CIN(1),N,0)                                 GRAPH      54
      CALL PLTDT2(1.0,TIM(1),STC(1),N,0)                                 GRAPH      55
      CALL PLTDT2 (1.0,TIM(1),DAMPLT(1),N,0)                             GRAPH      56
      CALL PLTDT2(1.0,TIM(1),TNR(1),N,0)                                 GRAPH      57
C     ---------------------------- STRAIN PLOTS -------------------------- GRAPH      58
      DO 100 I=1,NSTRN                                                   GRAPH      59
      J=1+MAXC*(I-1)                                                     GRAPH      60
      CALL STRAIN (TIM,EPSS1,EPSS2,          ETAG(I),PN(I),METAG(I),J,N) GRAPH      61
  100 CONTINUE                                                           GRAPH      62
```

```
C                    PLOTTING P(N).  9/30/75                          GRAPH    63
      IF(NNPF .EQ.0)GOTO 35                                           GRAPH    64
      YBAR = 40.0                                                     GRAPH    65
      DO 33 I=1,NNPF                                                  GRAPH    66
      YBAR = YBAR + 10.0                                              GRAPH    67
      IF(YBAR .GT. 30.0)CALL PLTPGF                                   GRAPH    68
      IF(YBAR .GT. 30.0)YBAR = 2.0                                    GRAPH    69
      ENCODE(7,30,STEP)NPF(I)                                         GRAPH    70
   30 FORMAT(2HP(,I3,2H)>)                                            GRAPH    71
      CALL PLTSCA(XBAR,YBAR,0.0,0.0,1.0,1.0)                          GRAPH    72
      CALL PLTSYM(.1,SYM2(1),0.0,     3.0,-0.6)                       GRAPH    73
      CALL PLTSYM(.1,STEP,90.0,    -1.2,2.5)                          GRAPH    74
      CALL FIXSCA(PD(1,I),N,YL,YS1,YMIN1,YMAX1,DY1)                   GRAPH    75
      CALL PLTSCA(XBAR,YBAR,XMIN1,YMIN1,XS1,YS1)                      GRAPH    76
      CALL PLTAXS(DX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,4)                  GRAPH    77
      CALL LABELA(DX1,DY1,XMIN1,XMAX1,YMIN1,YMAX1,XS,1.0)             GRAPH    78
      CALL PLTDT2(1.0,TIM(1),PD(1,I),N,0)                             GRAPH    79
   33 CONTINUE                                                        GRAPH    80
   35 CONTINUE                                                        GRAPH    81
      CALL PLTPGE                                                     GRAPH    82
      RETURN                                                          GRAPH    83
   50 FORMAT (F7.3,1H>)                                               GRAPH    84
      END                                                             GRAPH    85
      SUBROUTINE STRAIN (X,Y,Z,      ETA2,     PN1,NETA,J,N)          STRAIN    2
      DIMENSION X(1),Y(1),Z(1),        SYM2(3),SYM3(2),SYM4(2),SYM5(2), STRAIN    3
     1SYM6(2),X1(2),X2(2)                                             STRAIN    4
      LEVEL2,X,Y,Z                                                    STRAIN    5
      DATA I/0/,YS/100./,XS/1.0E6/                                    STRAIN    6
      DATA(SYM2(K),K=1,3)/10HETA2 =     ,10H       (   ,5H    )>/,'  STRAIN    7
     2     (SYM3(K),K=1,2)/10HTIME (MICR,10HOSECONDS)>/,              STRAIN    8
     3     (SYM4(K),K=1,2)/10HSTRAIN (%),1H>/,                        STRAIN    9
     4     (SYM5(K),K=1,2)/10HETA1 COMPO,5HNENT>/,                    STRAIN   10
     5     (SYM6(K),K=1,2)/10HETA2 COMPO,5HNENT>/                     STRAIN   11
C                                                                     STRAIN   12
      IF(I.EQ.0)GOTO 10                                               STRAIN   13
      I=I+1                                                           STRAIN   14
      YBARN=YBARN+10.0                                                STRAIN   15
      IF(I.LE.3)GOTO 25                                               STRAIN   16
      CALL PLTPGF                                                     STRAIN   17
      GOTO 20                                                         STRAIN   18
   10 CALL PLTPGF                                                     STRAIN   19
      XL=7.9                                                          STRAIN   20
      YL=6.4                                                          STRAIN   21
      CALL FIXSCA(X(1),N,XL,XS1,XMIN,XMAX,DX)                         STRAIN   22
   20 YBARN=0.0                                                       STRAIN   23
      I=1                                                             STRAIN   24
   25 YBAR=2.0+YBARN                                                  STRAIN   25
      XBAR=2.0                                                        STRAIN   26
      CALL PLTSCA (XBAR,YBAR,0.0,0.0,1.0,1.0)                         STRAIN   27
      CALL PLTSYM (.1,SYM4(I),90.0,    -1.2,2.5)                      STRAIN   28
      CALL PLTSYM (.1,SYM3(1),0.0,     3.0,-0.6)                      STRAIN   29
      CALL PLTSYM (.1,9HLOCATION>,0.0,     3.0,-1.0)                  STRAIN   30
      CALL PLTSYM (.3,2H(>,0.0,     3.8,-1.1)                         STRAIN   31
      CALL PLTSYM (.3,2H)>,0.0,     6.6,-1.1)                         STRAIN   32
      IF(NETA.NE.0)GOTO 30                                            STRAIN   33
      CALL PLTSYM (.1,6HOUTER>,0.0,     6.8,-1.0)                     STRAIN   34
      GOTO 35                                                         STRAIN   35
   30 CALL PLTSYM (.1,6HINNER>,0.0,     6.8,-1.0)                     STRAIN   36
   35 CALL PLTSYM (.1,SYM2(1),0.0,     4.1,-1.1)                      STRAIN   37
      ENCODE(8,50,STEP)ETA2                                           STRAIN   38
      CALL PLTSYM (.1,STEP,0.0,     4.8,-1.1)                         STRAIN   39
      ENCODE(8,50,STEP)PN1                                            STRAIN   40
      CALL PLTSYM (.1,STEP,0.0,     5.7,-1.1)                         STRAIN   41
```

```
      X1(1)=2.95                                                        STRAIN    42
      X2(1)=-1.3                                                        STRAIN    43
      X1(2)=3.576                                                       STRAIN    44
      X2(2)=-1.3                                                        STRAIN    45
      CALL PLTDTS (1.0,X1(1),X2(1),2.0)                                 STRAIN    46
      CALL PLTSYM (.1,SYM6(1),0.0,     3.7,-1.3)                        STRAIN    47
      X2(1)=-1.5                                                        STRAIN    48
      X2(2)=-1.5                                                        STRAIN    49
      CALL PLTDTS (4.0,X1(1),X2(1),2.0)                                 STRAIN    50
      CALL PLTSYM (.1,SYM6(1),0.0,     3.7,-1.5)                        STRAIN    51
      CALL FIXSCA (Y(J),N,YL,YS1,YMIN,YMAX,DY)                          STRAIN    52
      CALL CONSCA (Z(J),N,YL,YS1,YMIN,YMAX,DY)                          STRAIN    53
      CALL PLTSCA (XBAR,YBAR,XMIN,YMIN,XS1,YS1)                         STRAIN    54
      CALL PLTAXS (DX,DY,XMIN,XMAX,YMIN,YMAX,4)                         STRAIN    55
      CALL PLTDT2 (1.0,X(1),Y(J),N,0)                                   STRAIN    56
      CALL PLTDT2 (4.0,X(1),Z(J),N,0)                                   STRAIN    57
      CALL LABELA (DX,DY,XMIN,XMAX,YMIN,YMAX,XS,YS)                     STRAIN    58
      RETURN                                                            STRAIN    59
   50 FORMAT (F7.3,1H>)                                                 STRAIN    60
      END                                                               STRAIN    61
      SUBROUTINE FIXSCA (X,NPTS,SIZE,XSCALE,XMIN,XMAX,DX)               FIXSCA     2
      DIMENSION X(1),T(2)                                               FIXSCA     3
C           MUST READ IN FIXSCA TO PUT X IN LEVEL 2.                    FIXSCA     4
     LEVEL2,X                                                           FIXSCA     5
C(I)  X-ALINEAR ARRAY OF NUMBERS IN DATA UNITS                          FIXSCA     6
C(I)  NPTS-THE NUMBER OF X VALUES                                       FIXSCA     7
C(I)  THE LENGTH OF X DIMENSION OF THE GRAPH IN PLOTTER UNITS           FIXSCA     8
C(R)  XSCALE-THE SCALE IN DATA UNITS/PLOTTER UNIT                       FIXSCA     9
C(R)  XMIN-THE ADJUSTED MINIMUM IN DATA UNITS                           FIXSCA    10
C(R)  XMAX-THE ADJUSTED MINIMUM IN DATA UNITS                           FIXSCA    11
C(R)  DX-THE DELTA X FOR AXIS IN DATA UNITS                             FIXSCA    12
      LOGICAL CONT                                                      FIXSCA    13
      CONT=.FALSE.                                                      FIXSCA    14
      GOTO 100                                                          FIXSCA    15
      ENTRY CONSCA                                                      FIXSCA    16
      CONT=.TRUE.                                                       FIXSCA    17
  100 TXMI=X(1)                                                         FIXSCA    18
      TXMA=X(1)                                                         FIXSCA    19
      DO 120 I=1,NPTS                                                   FIXSCA    20
      IF(X(I).GE.TXMI)GOTO 110                                          FIXSCA    21
      TXMI=X(I)                                                         FIXSCA    22
      GOTO 120                                                          FIXSCA    23
  110 IF(X(I).LE.TXMA)GOTO 120                                          FIXSCA    24
      TXMA=X(I)                                                         FIXSCA    25
  120 CONTINUE                                                          FIXSCA    26
      IF(.NOT.CONT)GOTO 140                                             FIXSCA    27
      IF(TXMI.LT.XMIN.OR.TXMA.GT.XMAX)GOTO 125                          FIXSCA    28
      TXMI=XMIN+ABS(XMIN)*0.000001                                      FIXSCA    29
      TXMA=XMAX-ABS(XMAX)*0.000001                                      FIXSCA    30
      GOTO 140                                                          FIXSCA    31
  125 IF(TXMI.LT.XMIN)GOTO 130                                          FIXSCA    32
      TXMI=XMIN+ABS(XMIN)*0.000001                                      FIXSCA    33
  130 IF(TXMA.GT.XMAX)GOTO 140                                          FIXSCA    34
      TXMA=XMAX-ABS(XMAX)*0.000001                                      FIXSCA    35
  140 DIFF=TXMA-TXMI                                                    FIXSCA    36
      IF(DIFF.LT.0.000001) DIFF=0.000001                                FIXSCA    37
      ENCODE (20, 150, T) DIFF                                          FIXSCA    38
  150 FORMAT(1PE12.5)                                                   FIXSCA    39
      DECODE (20, 160, T) COFF, IEXP                                    FIXSCA    40
  160 FORMAT(F8.5,1X,I3)                                                FIXSCA    41
  170 IF(COFF.GT.2.0)GOTO 180                                           FIXSCA    42
      DELTA=0.1                                                         FIXSCA    43
      GOTO 200                                                          FIXSCA    44
```

```
180 IF(COEF.GT.4.0)GOTO 190                                    FIXSCA     45
    DELTA=0.2                                                  FIXSCA     46
    GOTO 200                                                   FIXSCA     47
190 DELTA=0.5                                                  FIXSCA     48
200 DELTA=DELTA*10.0**IEXP                                     FIXSCA     49
    DX=DELTA                                                   FIXSCA     50
    XMIN=AINT(ABS(TXMI)/DELTA)*DELTA                           FIXSCA     51
    IF(TXMI.LT.0)XMIN=-(XMIN+DELTA)                            FIXSCA     52
    XMAX=AINT(ABS(TXMA)/DELTA)*DELTA                           FIXSCA     53
    IF(TXMA.LT.0)GOTO 210                                      FIXSCA     54
    XMAX=XMAX+DELTA                                            FIXSCA     55
    GOTO 220                                                   FIXSCA     56
210 XMAX=-XMAX                                                 FIXSCA     57
220 XSCALE=(XMAX-XMIN)/SIZE                                    FIXSCA     58
    RETURN                                                     FIXSCA     59
    END                                                        FIXSCA     60
```

APPENDIX E

EXAMPLES

Two examples are given in this appendix. These were originally jobs run on the BRLESC computer, now defunct, and transferred to the CDC CYBER 70 Model 76 computer to check the program there. The two jobs test, and hence illustrate, many of the available options and internal variations of RPSL1D. For both examples, the complete SCOPE 2.1 batch job is listed and a sample of the output is given. The sections of the batch job are separated in the listings by question marks which listed for the multipunch end of record and end of information signals.

E.1 Example 1: Pressure Loaded Flat Plate

The first example involves finding the response along the shorter symmetry line of a flat 4 by 15 inch soft aluminum plate. (Results from the standard REPSIL and the slab symmetric version of RPSL1D are very similar for this elongated plate.) This example demonstrates the introduction of a new PRESS subroutine, the use of the optional coding PLOTP, and the activating of Gaussian integration and damping. Both the new PRESS routine and the option PLOTP are inserted by making changes in the program through UPDATE; both also require input data cards. With the catalogued INGEOM, Gaussian integration is automatically used with slab symmetry if IB = 0 on input card 14. Damping was activated from time Δt on by setting MDAMP = 0 on input card 5. The damping option was used here to help simulate displacement of the plate by quasi-static pressure loading. This is not the usual use of damping. Damping is a device normally used to bring a responding surface to rest at its final equilibrium configuration after the loading is removed. Since subroutine DAMP is not entered until cycle one, when NCYCLE = 1, the subroutine did not shut off pressure loading as it will if entered with NCYCLE = MDAMP. This run terminated at cycle 3546 with a displacement of .66 inches.

A listing of the batch job deck and samples of output for Example 1 follow. The first part of the job deck is the SCOPE 2.1 control statements to set up and run RPSL1D and the plotting. The next set of cards is UPDATE directives and new FORTRAN statements for RPSL1D. The input data cards for RPSL1D are next. (The input and output for this example are in the now forbidden pound-inch-second system. If this is discomforting, pretend the units are SI units for some very exotic material.) After the data there are UPDATE changes for the plotting program, including changes for the PLOTP option and a change to permit up to 6000 time cycles. Finally, there is a data card to control plotting.

Following the job listing are sample listings of output. This starts with a summary of the input and the initial Cartesian coordinates as prescribed by INGEOM. All the output should be inspected to uncover errors, but these two sets should receive very close attention.

The output at cycle 3000 for this example is listed next. This includes:

- Displacement increments, coordinates, and pressure at each mesh point including the virtual external point at N=1 and the symmetry point at N=13.

- The LMAT matrix. A "1" signifies plasticity, and a "0" denotes an elastic response, on the current cycle. For each column labeled N, there are two rows for each integration station K. The first row is for mesh point N. The second row is for midmesh N.

- The components of the surface normal at each mesh point, the 2 columns on the left, and at each midmesh.

- Surface strains at all mesh points. For each mesh point, the strains on the upper surface are listed first, then the strains on the lower surface are listed. EC1 and EC2 are the elongation strains in the ξ^1 and ξ^2 directions, respectively. E11 and E22 are internal covariant components of strain. For slab symmetry, EC1 and E11 are zero.

- The tabulated output for the NSTRN prescribe surface strains accompanied by prints of extreme strains and maximum deflection for the current cycle, and for the entire run.

- The energy balance summary.

In our listing, this is followed by three prints noting the use of the kinetic energy annihilating procedure (a part of damping which stops motion at kinetic energy maxima) and finally a print of the prescribed surface strains, etc. at cycle 3500.

The occurrence and frequency of this output is controlled by input cards 8, 9, and 10.

Figures E.1.1 through E.1.4 are samples of plotting output. For most publication purposes, it is desireable to do some relabeling, but these plots have not been retouched. Figure E.1.1 shows displacement at selected time steps. Figure E.1.2 shows the displacement of the selected central mesh point and the energy balance as functions of time. The scallops in these plots occur when the stopping procedure was exercised. This energy balance plot shows the four possible curves. These are, from low to high: kinetic energy, kinetic energy plus elastic strain energy, kinetic energy plus elastic strain energy plus the work in damping, and the total energy. Figure E.1.3 is the first two of the six strain plots requested for this run. Figure E.1.4 is a plot of force per initial unit area at mesh point 3. The pressure increased linearly to a maximum of 80 psi and then remained constant, but the area increased throughout the entire run.

Listing of the Batch Job Cards for Example 1

```
ROUTE,STMT7,TAG.    DUPLICATE RUN OF 1/3/77.  TEST GAUSS, PLOTP, LINPRS.
ACCOUNT,*****,                    *PETMAN,P30S,A3S79
ATTACH(OLDPL,PPSLIB,ID=JOW)
UPDATE,P.
FTN(A,I,SL=0,R=0)
MAP,OFF.
LGO.
EXIT(U)             DATA TO PLOT TAPE
REWIND(PLOT)
RETURN,COMPILE.
RETURN,LGO.
RETURN,OLDPL.
ATTACH(OLDPL,PPSLIB*PLOTLIB,ID=JOW)
BEGIN,ATTACH,PLOTLIB.
UPDATE(F)
FTN(A,I,FL=F,SL=0,R=0,LCM=I)
MAP,OFF.
LGO.
EXIT(U)
BEGIN,PLOT,CALCOMP,TAPE13.
?
*IDENT  PLOTP                                                          PLOTP
*I  PPSLIB,S
C                    PLOTTING P(N).  9/30/75                           PLOTP
*I  MAIN,2R
      COMMON NNPF,NPF( 9),PDAT( 9)                                     PLOTP
*I  START,24
      READ(5,110)NNPF,(NPF(I),I=1,NNPF)                                PLOTP
*I  PDATA,13
      WRITE(PLOT)NNPF,(NPF(I),I=1,NNPF)                                PLOTP
*I  PDATA,19
      DO 24 I=1,NNPF                                                   PLOTP
      J=NPF(I)                                                         PLOTP
   24 PDAT(I)=P(J)                                                     PLOTP
*I  PDATA,27
     1 ,(PDAT(J),J=1,NNPF)                                             PLOTP
*I  PDATA,42
      DO 44 I=1,NNPF                                                   PLOTP
      J=NPF(I)                                                         PLOTP
   44 PDAT(I)=P(J)                                                     PLOTP
*I  PDATA,45
     1 ,(PDAT(J),J=1,NNPF)                                             PLOTP
*IDENT  LINPRS
*I  PPRESS,4,PPRESS,14
C         PRESSURE BY LINEAR INTERPOLATION WITH RESPECT TO TIME IN      LINPRS
C         INPUT TABLE TPR(I),PPR(I).    10/7/76                         LINPRS
C         FORMAT (2F10.3)          TERMINATE WITH NEGATIVE TPR.         LINPRS
      DIMENSION TPR(51),PPR(51)                                        LINPRS
      DATA IPRESS/0/                                                   LINPRS
      IF(IPRESS .GT. 0)GOTO 4                                          LINPRS
      WRITE(6,100)                                                     LINPRS
      I = 0                                                            LINPRS
    2 I = I+1                                                          LINPRS
      READ( 5,101)TPR(I),PPR(I)                                        LINPRS
      IF(TPR(I) .LT. 0.0)GOTO 3                                        LINPRS
```

```
      WRITE(6,102)I,T=>(I),PPP(I)                                          LINPRS
      GOTO 2                                                               LINPRS
    3 IPRESS = I-1                                                         LINPRS
    4 CALL INTERP(TIME,P0,TPP,PPP,IPRESS,2)                                LINPRS
      DO 5 N=1V,N2V                                                        LINPRS
      P(N) = P0                                                            LINPRS
    5 CONTINUE                                                             LINPRS
      RETURN                                                               LINPRS
  100 FORMAT(//' PRESSURE FROM A TABLE'/4X,'I',5X,'TIME',9X,'PRESSURE')   LINPRS
  101 FORMAT(2F10.3)                                                       LINPRS
  102 FORMAT(I5,2F15.7)                                                    LINPRS
?
   STRUCT, MODEL,  .020 PLATE,    GRADUALLY INCREASING PRESSURE TO 40PSI 1/3/77
    5   10    4 2.0
 4000     0 0000 0.500000E-6
    2    2    1    1
    1 0000    0 0.1            0.001
10.000000E6 0.33        5P00.       2.530000E-4 .020          4    0
5P00.           .0005P
6700.           .035
11P90.          .10
13050.          .25
 500    1    1    1
   6  500 1000 1500 2000 3000 3545
   6  500 1000 1500 2000 3000 3545
   2  500 3545
   1   3
0.0        2.0          6
0.0        0.0        45.0       135.0        0
0.0        1.0        45.0       135.0        0
0.0        2.0        45.0       135.0        0
0.0        0.0        45.0       135.0        1
0.0        1.0        45.0       135.0        1
0.0        2.0        45.0       135.0        1
2.0        7.5           0
0.0        0.0
0.00135    P0.0
1.0        P0.0
-1.0
?
*TFRAT PLOTP
*C  PPIPLT.12
      READ(NPLOT)NAPP,(NPP(I),I=1,NAPP)                                    PLOTP
*D  PPIPLT.40
     1 ,(PPAT(J),J=1,NUPP)                                                 PLOTP
*TC  P6002
*C  MAIN.7,MAIN.9
     1  TIM( 6002),    U2( 6002),    U3( 6002),  CIN( 6002),  STC( 6002), 6002
     2  TMP( 6002),DAMPLT( 6002),EPSS1(36012),EPSS2(36012)                  6002
      COMMON/PLOTP/P0( 6002,9)                                             6002
*D  PPIPLT.10
      MAX( = 6002                                                          6002
?
 1.0        3.0
?
```

```
                                    BRL REPSIL CODE

    STRUCT. MODEL.  .020 PLATE.    GRADUALLY INCREASING PRESSURE TO REPSI 1/3/77

                        COMPUTATIONS FOR ENERGY USE DETAI= .150000E+02
                    10 MESHES IN ETA2 DIRECTION    (DETAP= .200000E+00)

                            PENDING  TIME INCREMENT=  .1644A0E-04
                            PERMANE  TIME INCREMENT=  .9496ABE-06
                            INPUT    TIME INCREMENT=  .000000E-06

                          TIME INCREMENT USED BY REPSIL=  .000000E-06

          YOUNG'S MODULUS = .100000E+08
          POISSON'S RATIO = .330000E+00          YIELD STRESS     = .500000E+04
          MASS DENSITY    = .253000E-03          THICKNESS        = .200000E-01

                                START AT TIME STEP     0
                                FINAL  TIME  STEP  4000
                                SURFACE STRAINS EVERY  500 TIME STEP
                                RESTART  WRITE  EVERY 9999 TIME STEP

                          LAYER =    4                  NSTRN =    6
                          LOAD  =    1                  LPRESS = 9999

                                BOUNDARY CONDITIONS
    1/2/3/4/     = CLAMPED/SYMMETRY/HINGED/FREE/
                          END1 (IBCE1) =   1
                          END2 (IBCE2) =   2

                          PRINT OPTION CONTROL CARD
                            0/1 = NO PRINT/PRINT
                              1 DISPLACEMENT INCREMENTS
                              1 CARTESIAN COORDINATES, PRESSURE
                              1 SURFACE NORMAL VECTOR COMPONENTS

    PRINT INFORMATION AT THE FOLLOWING TIME STEPS
      500 1000 1500 2000 3000 3545
    PRINT L MATRIX (LMAT) AT THE FOLLOWING TIME STEPS
      500 1000 1500 2000 3000 3545
    3-D PLOTS FOR THE FOLLOWING TIME STEPS
      500 3545

              CONSTITUTIVE RELATION    ELASTOPLASTIC-WORK HARDENING-STRAIN RATE INDEPENDENT
            STRESS-STRAIN APPROXIMATION HAS  4 SUBLAYERS

                        STRESS-STRAIN AND STRAIN RATE PARAMETERS
          J        SSI(J)          SEPS(J)          PSR(J)          1/PSR(J)
          1    5.000000E+03  5.000000E-04        0.              0.
          2    9.700000E+03  3.500000E-02        0.              0.
          3    1.180000E+04  1.000000E-01        0.              0.
          4    1.305000E+04  2.500000E-01        0.              0.

                        START DAMPING AFTER TIME STEP    0     TIME =0.
                  DAMPP = .100E+00                 DFACT = .100E-02

    PRESSURE FROM A TABLE
        I      TIME          PRESSURE
        1   0.               0.
        2   .11E-00E-02      .0000000E+02
        3   .1000000E+01     .0000000E+02
```

Selected Output from Example 1

	INITIAL COORDINATES			PRESSURE
N	R(N)	Z(N)		P(N)
1	0.	-.19999999999999993E+00	0.	
2	0.	0.	0.	
3	0.	.20000000000000025E+00	0.	
4	0.	.40000000000000044E+00	0.	
5	0.	.60000000000000142E+00	0.	
6	0.	.80000000000000071E+00	0.	
7	0.	.10000000000000000E+01	0.	
8	0.	.12000000000000036E+01	0.	
9	0.	.13999999999999999E+01	0.	
10	0.	.16000000000000018E+01	0.	
11	0.	.18000000000000044E+01	0.	
12	0.	.20000000000000000E+01	0.	
13	0.	.21999999999999956E+01	0.	

TIME STEP 3000 TIME .27000000E-02

DISPLACEMENT INCREMENTS BETWEEN T.S.2999 AND3000 — COORDINATES — PRESSURE

N	DR(N)	DZ(N)	R(N)	Z(N)	P(N)
1	-.90524320739055504E-05	.31092604190449935E-05	-.11930596603739751E+00	-.17646999551405081E+00	0.
2	0.	0.	0.	0.	0.
3	-.90524320739055504E-05	-.31092604190449949E-05	-.11930596603739751E+00	.17646999551405908E+00	.80000000000000000E+02
4	-.17943796663594437E-04	-.57647304741271105E-05	-.22690209908512356E+00	.34047220744069261E+00	.80000000000000000E+02
5	-.30000480122557748E-04	-.76150646147752215E-05	-.32205861267523691E+00	.55119767651540561E+00	.80000000000000000E+02
6	-.39662901239764808E-04	-.86500290708447407E-05	-.40460674672805171E+00	.74772209947347779E+00	.80000000000000000E+02
7	-.40517243044700279E-04	-.88399990532566208E-05	-.47467057174709491E+00	.94914312622366674E+00	.80000000000000000E+02
8	-.56390250994179913E-04	-.82156800490057373E-05	-.53197230144426655E+00	.11545054570602508E+01	.80000000000000000E+02
9	-.62813061646480293E-04	-.68652595081449408E-05	-.57653071270083460E+00	.13632021556634308E+01	.80000000000000000E+02
10	-.67591481202084922E-04	-.49252066590952000E-05	-.60036894534624971E+00	.15741720730062128E+01	.80000000000000000E+02
11	-.70535767391029818E-04	-.25690045374170083E-05	-.62746434735104731E+00	.17866995269297668E+01	.80000000000000000E+02
12	-.71530780241909808E-04	0.	-.63382086651530331E+00	.20000000000000000E+01	.80000000000000000E+02
13	-.70535767391029818E-04	.25690045374170083E-05	-.62746434735104731E+00	.22133004730703058E+01	0.

TIME STEP 3000 TIME 2.7000000E-03. SUBDIVISIONS OF TIME INCREMENT IN STRESS

LMAT(N,K)

K	N= 2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	0
3	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	0
4	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	0

TIME STEP 3000 TIME .27000000E-02

SURFACE NORMAL VECTOR COMPONENTS

N	SNR(N)	SNZ(N)	SNRR(N)	SNZR(N)
2	.10000000000000000E+01	0.	.87826336334062840E+00	.56033900537770062E+00
3	.84629083411740517E+00	.53270922436732708E+00	.86340797440677971E+00	.50450636247845455E+00
4	.87759146642408292E+00	.47972908140906267E+00	.89408130861708096E+00	.44643941170271753E+00
5	.90808231082323094E+00	.41722697225304437E+00	.92180340744020730E+00	.38750820408161043E+00
6	.93360791663089947E+00	.35800700255332108E+00	.94459521760366598E+00	.32823754253014058E+00
7	.95430593113055343E+00	.29857577593196089E+00	.96327436645840871 25E+00	.26862925080973350E+00
8	.97104866013602308E+00	.23800176910306941E+00	.97793395704073078E+00	.20691427082363630E+00
9	.98378253012345808E+00	.17913061330219280E+00	.98880904708290395E+00	.14910866040276030E+00
10	.99208474247110011E+00	.11939000801706196E+00	.99590871814039370E+00	.89400952908205705E-01
11	.99821739955227064E+00	.59602795507023564 5E-01	.99955513556657750E+00	.29025008565771662 5E-01
12	.10000000000000000E+01	0.	.10000000000000000E+01	0.

SURFACE STRAINS. UPPER/LOWER. (EC1,EC2,E11,E22)

MAXIMUM DEFLECTION AT TIME= 2.700000E-03, CYCLE= 3000
DEFLECTION= 6.330209E-01 (N = 12)
MAXIMUM DEFLECTION BETWEEN CYCLES 0 AND 3000
DEFLECTION= 6.330209E-01 (N = 12) TIME= 2.700000E-03

TIME STEP 3000 TIME= .27000000E-02 KINETIC= .47062437E+00 ELASTIC= .89406126E+01 PLASTIC= .81497066E+03
TOTAL ENERGY= .15727618E+04

STOPPING PROCEDURE ACYCLE,CINES,CINER,CINET,TDAMP,CINEP
3003 6.7057000E-01 6.7058696E-01 6.7057048E-01 7.5007731E+02 9.5595603E-01

STOPPING PROCEDURE NCYCLE,CINES,CINER,CINET,TDAMP,CINEP
3230 4.0156321E-01 4.0158593E-01 4.0157457E-01 7.5303760E+02 6.7057048E-01

STOPPING PROCEDURE ACYCLE,CINES,CINER,CINET,TDAMP,CINEP
3392 3.4070542E-01 3.4070020E-01 3.4070604E-01 7.9455529E+02 4.0157457E-01

TIME STEP 3500 TIME= .31500000E-02

SURFACE STRAINS / STRAIN GAGE READING

FIAZ	N	FACE	ANGLE 0	ANGLE 90	ANGLE		ANGLE	
0.000	2.000	OUTER	0.	.12654681E+00	45.00	.65154380E-01	135.00	.65154380E-01
1.000	7.000	OUTER	0.	.68055390E-01	45.00	.34587430E-01	135.00	.34587430E-01
2.000	12.000	OUTER	0.	.68559107E-01	45.00	.34847460E-01	135.00	.34847460E-01
0.000	2.000	INNER	0.	.11564169E-01	45.00	.57907044E-02	135.00	.57907044E-02
1.000	7.000	INNER	0.	.74400090E-01	45.00	.37912400E-01	135.00	.37912400E-01
2.000	12.000	INNER	0.	.74032761E-01	45.00	.38090900E-01	135.00	.38090900E-01

EXTREME STRAINS AT TIME= 3.150000E-03, TIME CYCLE= 3500

	STRAIN	N
EPSU1MAX	0.00000000	2
EPSU2MAX	.12654681	2
EPSL1MAX	0.00000000	2
EPSL2MAX	.07403276	12
EPSU1MIN	0.00000000	2
EPSU2MIN	.00741600	3
EPSL1MIN	0.00000000	2
EPSL2MIN	.01156417	2

EXTREME STRAINS BETWEEN CYCLES 0 AND 3500

	STRAIN	TIME	N
EPSU1MAX	0.00000000	9.000000E-07	2
EPSU2MAX	.12654681	3.150000E-03	2
EPSL1MAX	0.00000000	9.000000E-07	2
EPSL2MAX	.07403276	3.150000E-03	12
EPSU1MIN	0.00000000	9.000000E-07	2
EPSU2MIN	-.00041305	3.700000E-04	5
EPSL1MIN	0.00000000	9.000000E-07	2
EPSL2MIN	-.01037070	1.023300E-03	2

MAXIMUM DEFLECTION AT TIME= 3.150000E-03, CYCLE= 3500
DEFLECTION= 6.595600E-01 (N = 12)
MAXIMUM DEFLECTION BETWEEN CYCLES 0 AND 3500
DEFLECTION= 6.595600E-01 (N = 12) TIME= 3.150000E-03

2	0.	1.1993422625159E-01	0.	1.2712633554488E-01	
	0.	7.7106425234284E-03	0.	7.7403695274023E-03	
3	0.	8.2125728268187E-02	0.	6.4855531345258E-02	
	0.	8.8723052986222E-02	0.	7.1884481986598E-02	
4	0.	6.2336485694378E-02	0.	6.4279484428854E-02	
	0.	6.8038107854878E-02	0.	7.1314397212184E-02	
5	0.	6.2591243884858E-02	0.	6.4958874855974E-02	
	0.	6.9867561847574E-02	0.	7.1452725842188E-02	
6	0.	6.2876237813784E-02	0.	6.4852947643088E-02	
	0.	6.9232284856825E-02	0.	7.1628839178869E-02	
7	0.	6.3165241437495E-02	0.	6.5160165388425E-02	
	0.	6.9418995884889E-02	0.	7.1818938959678E-02	
8	0.	6.3435853024838E-02	0.	6.5447854888184E-02	
	0.	6.9584618439677E-02	0.	7.2885628881374E-02	
9	0.	6.3665978889198E-02	0.	6.5692657493338E-02	
	0.	6.9736961498688E-02	0.	7.2168858335988E-02	
10	0.	6.3842328554918E-02	0.	6.5888258812578E-02	
	0.	6.9855133997146E-02	0.	7.2295883549875E-02	
11	0.	6.3952753843672E-02	0.	6.5997731285767E-02	
	0.	6.9929827838521E-02	0.	7.2374917393339E-02	
12	0.	6.3998344795567E-02	0.	6.6037726878781E-02	
	0.	6.9955369195561E-02	0.	7.2402246835288E-02	

SURFACE STRAINS. UPPER/LOWER. (EC1,EC2,E11,E22)

TIME STEP 3000 TIME= .27000000E-02

SURFACE STRAINS					STRAIN GAGE READING				
ETA2	N	FACE	ANGLE 0	ANGLE 90	ANGLE		ANGLE		
0.000	2.000	OUTER	0.	.11993423E+00	45.00	.61602863E-01	135.00	.61602863E-01	
1.000	7.000	OUTER	0.	.63165241E-01	45.00	.32065969E-01	135.00	.32065969E-01	
2.000	12.000	OUTER	0.	.63990345E-01	45.00	.32491038E-01	135.00	.32491038E-01	
0.000	2.000	INNER	0.	.77106475E-02	45.00	.38627244E-02	135.00	.38627244E-02	
1.000	7.000	INNER	0.	.69410996E-01	45.00	.35207370E-01	135.00	.35207370E-01	
2.000	12.000	INNER	0.	.69955369E-01	45.00	.35568562E-01	135.00	.35568562E-01	

EXTREME STRAINS AT TIME= 2.700000E-03. TIME CYCLE= 3000

	STRAIN	N
EPSU1MAX	0.00000000	2
EPSU2MAX	.11993423	2
EPSL1MAX	0.00000000	2
EPSL2MAX	.06995537	12
EPSU1MIN	0.00000000	2
EPSU2MIN	.06712573	3
EPSL1MIN	0.00000000	2
EPSL2MIN	.00771064	2

EXTREME STRAINS BETWEEN CYCLES 0 AND 3000

	STRAIN	TIME	N
EPSU1MAX	0.00000000	0.0000000E-01	2
EPSU2MAX	.11993423	2.7000000E-03	2
EPSL1MAX	0.00000000	0.0000000E-01	2
EPSL2MAX	.06995537	2.7000000E-03	12
EPSU1MIN	0.00000000	0.0000000E-01	2
EPSU2MIN	-.00041305	3.7000000E-04	5
EPSL1MIN	0.00000000	0.0000000E-01	2
EPSL2MIN	-.01037070	1.0733000E-03	2

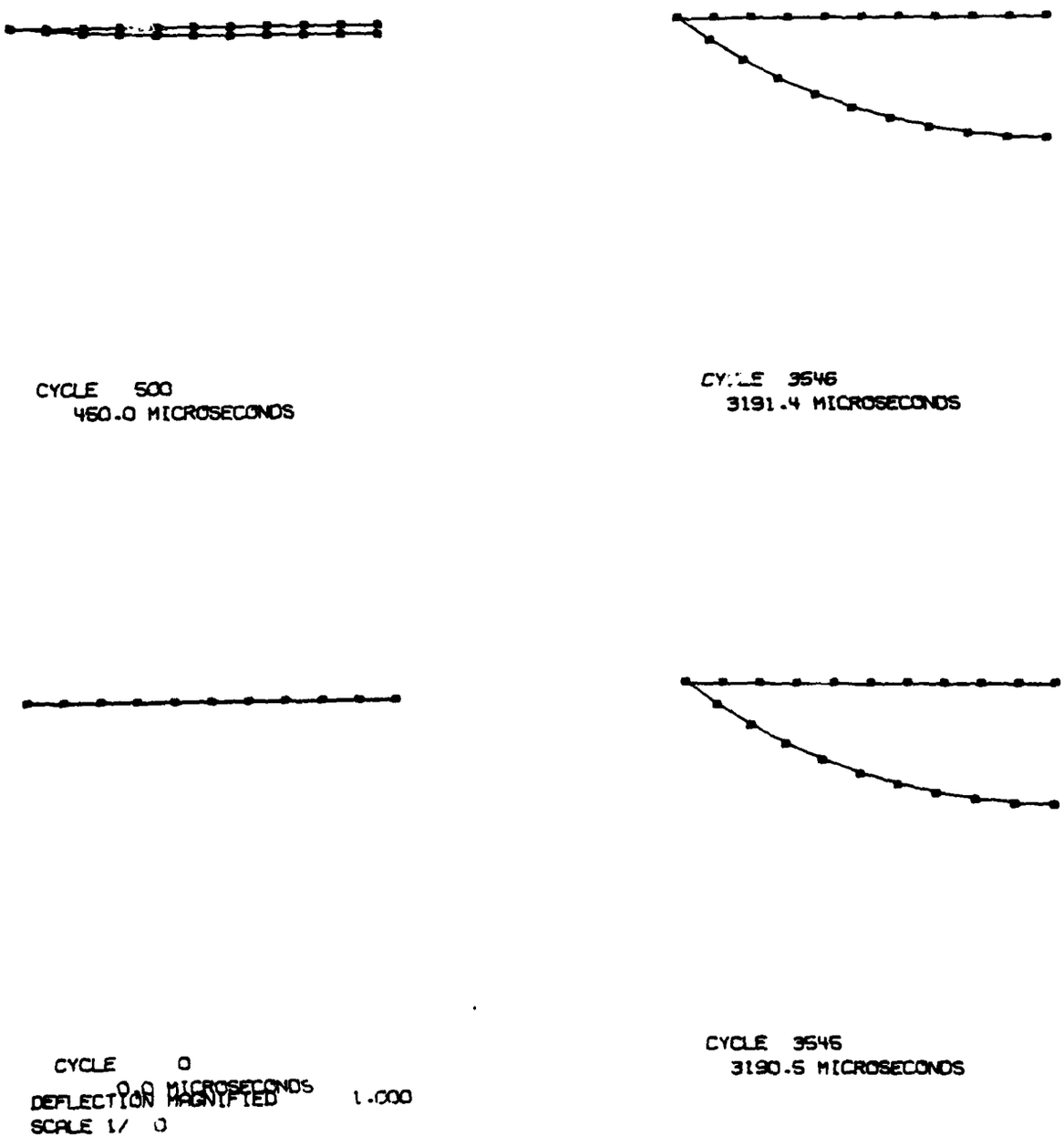


Figure E.1.1. Plots of the reference curve at selected times.

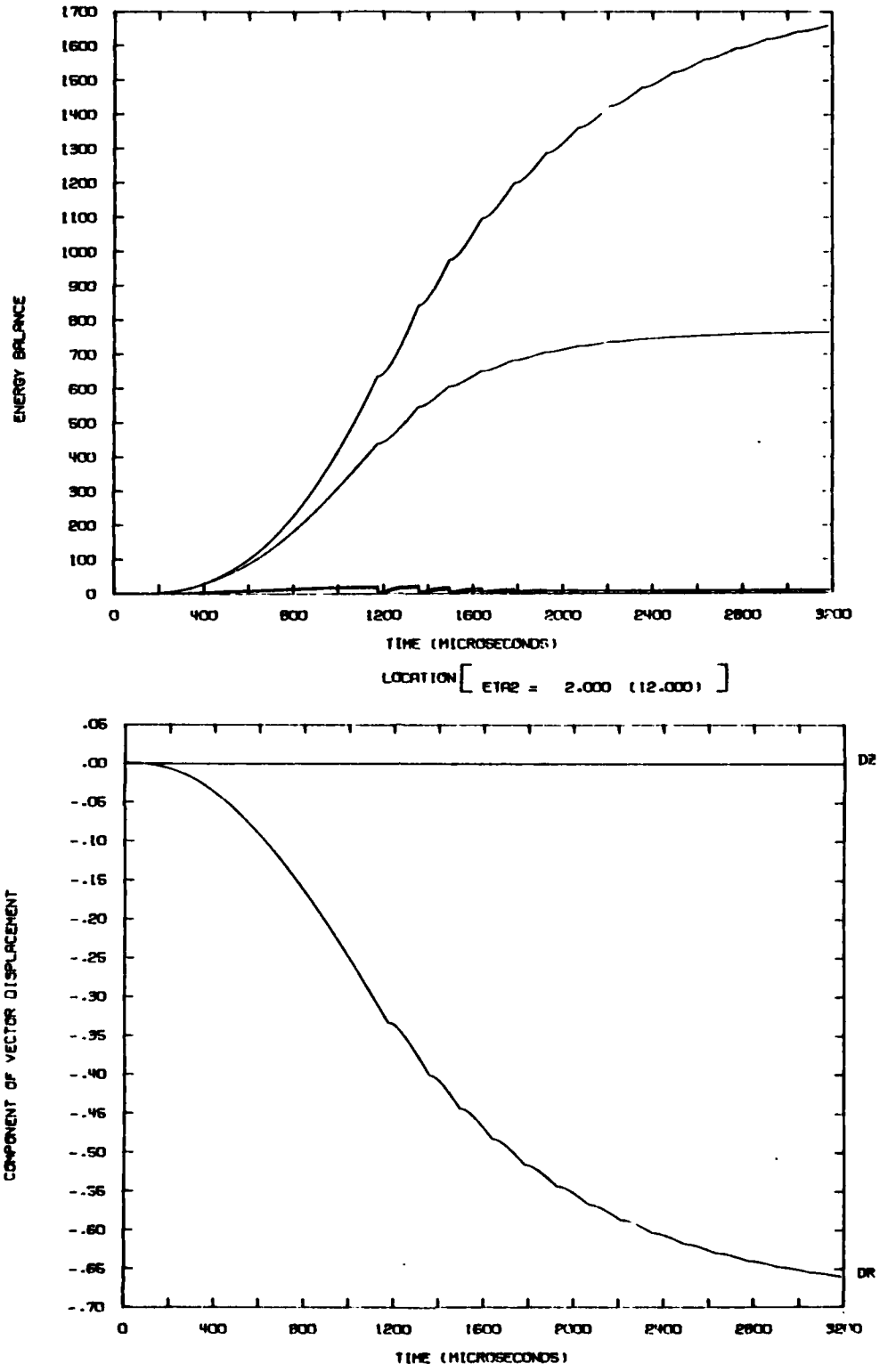


Figure E.1.2. Displacement history of the central point and energy balance curves.

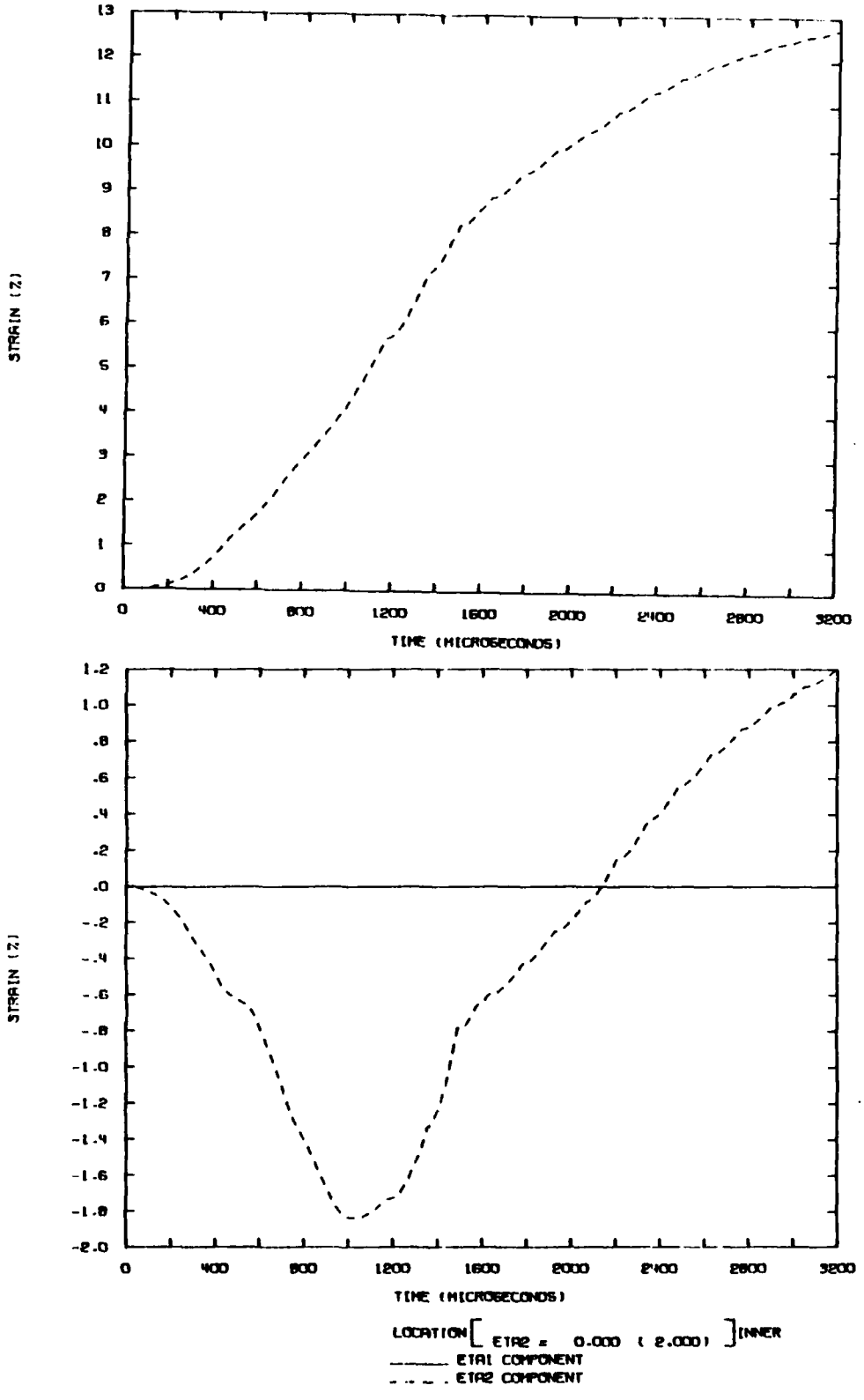


Figure E.1.3. Strain histories at the edge on the top and bottom surfaces

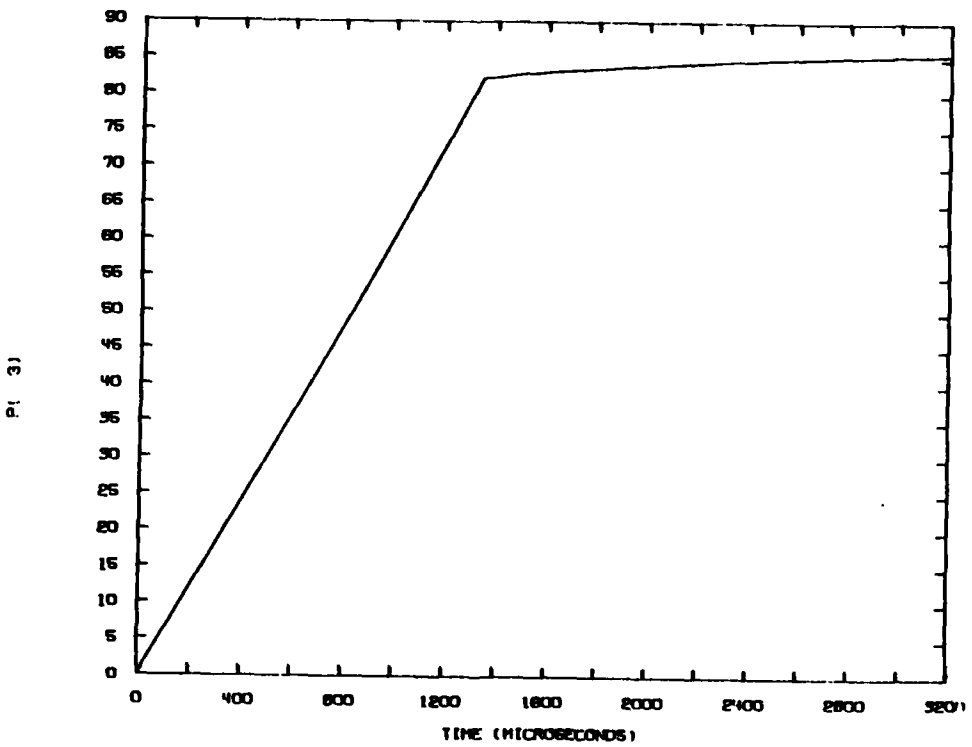


Figure E.1.4. Force per initial unit area at mesh point B.

E.2 Example 2: Eroding Rod

This example uses the beam option of RPSL1D and utilizes all seven of the options listed in Section 5. This program was designed to illustrate that the test strain-time records at two locations along a steel rod penetrating a steel plate could be reproduced by varying the force and the rate of erosion on the end. Two quite different tables of erosion and force were used, and both returned strains that were close to the test values at three locations. This example used a fast erosion rate. Mesh points 2 and 3 were eroded by time cycle 5.

The batch job cards and some sample output are listed. The batch job for this example is rather long. The first section of the run stream is the SCOPE 2.1 control statements. The next section is the UPDATE directives and new FORTRAN statements for RPSL1D. Breaking this section into the nine groups with different UPDATE card identifiers we have: PLOTP, SHR3/1, MSQSVS, BSTRS, APLFRC (including a subroutine ENDFRC which is adjusted for erosion), EAPFRC, 11/2/76 (a modification for EAPFRC), INVELC (an insert into INVEL which creates a uniform initial velocity toward end 1), and ERODE.

The next section in the job cards is the input data. The input, and output, for this example are in SI units: meters, seconds, kilograms, Pascals, etc. This leads to some awkward scaling on both input and output, but assures a consistent, unambiguous system. Along with the usual input data, this section contains a long table of strain vs. stress for the BSTRS option, input to approximate a solid cylindrical beam for INGEOM, no input for PRESS, a card with initial velocity for INVEL, three tables for ENDFRC, and a table for ERODE.

The next section is UPDATE changes for the file RPSL1DPLOT for options MSQSVS and PLOTP. Finally, there is an input data card to control the plotting.

The tabular output listed is two of the seven pages which reflect input (most of the omitted tabulations are simply copies of input tables), and the output at cycle 130. The output at cycle 130 includes:

- The displacement and coordinates. R and DR are effectively zero.
- The output from SHR3/1. The moment, MS, and shear, VS, are zero. Only axial force, QS, is significant.
- The table of plastic stress occurrence.
- The components of the unit normal vector. The rod has not bent.
- The surface strains.
- The energy balance.

Six of the output plots are shown next. (An isometric plot of the rod and plots of MS, QS, and VS were made at cycles 70 and 130. The isometric plot is uninteresting, and MS and VS are zero.) A plot of QS vs. mesh point number at cycle 130 is included to illustrate output of the MSQSVS option. We also include plots of several functions of time: the displacement of mesh point 5, strain plots at 19.26 mm and 39.67 mm, a plot of force on end 1, EFZ1, which was stored in P(53), and the erosion at end 1 which was stored in P(54).

Listing of the batch job cards for Example 2

```
WORTM,STMF7,T30.   DUPLICATE RUN OF 6/8/77 AND 10/12/77.   ERODING BOD.
ACCOUNT.           WORTMAN,B309,X3979
ATTACH,OLDPL,HPSL1D,ID=JDW.
UPDATE,F.
FTN(A,I,SL=0,R=0)
MAP,OFF.
LGO.
EXIT(U)            DATA TO PLOT TAPE
REWIND(PLOT)
RETURN,COMPILE.
RETURN,LGO.
RETURN,OLDPL.
ATTACH(OLDPL,HPSL1DPLOT,ID=JDW)
BEGIN,ATTACH,PLOTLIB.
UPDATE(F)
FTN(A,I,EL=F,SL=0,R=0)
MAP,OFF.
LGO.
EXIT(U)
BEGIN,PLOT,CALCOMP,TAPE13.
?
*IDENT  PLOTP
*I  HPSL1D,5
C                    PLOTTING P(N).  9/30/75                                PLOTP
*I  MAIN,28
      COMMON NNPE,NPE( 9),PDAT( 9)                                          PLOTP
*I  START,24
      READ(5,110)NNPE,(NPE(I),I=1,NNPE)                                     PLOTP
*I  PDATA,13
      WRITE(NPLOT)NNPE,(NPE(I),I=1,NNPE)                                    PLOTP
*I  PDATA,19
      DO 26 I=1,NNPE                                                        PLOTP
      J=NPE(I)                                                              PLOTP
   26 PDAT(I)=P(J)                                                          PLOTP
*I  PDATA,27
     1 ,(PDAT(J),J=1,NNPE)                                                  PLOTP
*I  PDATA,42
      DO 46 I=1,NNPE                                                        PLOTP
      J=NPE(I)                                                              PLOTP
   46 PDAT(I)=P(J)                                                          PLOTP
*I  PDATA,45
     1 ,(PDAT(J),J=1,NNPE)                                                  PLOTP
*IDENT  SHR3/1
*I  HPSL1D,5
C         ESTIMATE SHEAR FORCE, VS, AND MS AND QS.  3/1/76 (FOR A BEAM)  SHR 3/1
*I  MAIN,28
      COMMON AMS(103),QS(103),VS(103)                                       SHR 3/1
*I  RESULT,48
C         MS AT FREE END IS ZERO (4/5/76).   CHANGE WITH APLFRC.          SHR 3/1
      AMS(N2P) = 0.0                                                        SHR 3/1
*I  RESULT,54
      AMS(N) = A22*F22                                                      SHR 3/1
*R  RESULT,62
      VS(N) = (AMS(N+1) - AMS(N))/(DELT32*SRJ)                              SHR 3/1
  209 QS(N) = A22*(C22-RM22*F22)                                            SHR 3/1
```

```
C         VS,MS,QS PRINT CONTROLLED BY CODING.   SHR 3/1/76.
C         TO PRINT EVERY 10 CYCLES.
C       IF(MOD(NCYCLE,10) .EQ. 0) GOTO 212                                  9/15/76
        IF(NCYCLE .EQ. 70 .OR. NCYCLE .EQ. 130)GOTO 212                     TEST
        GOTO 218                                                           SHR 3/1
  210 FORMAT(//' CYCLE  N',6X,'MS',13X,'VS',13X,'QS',13X,'A22',12X,        SHR 3/1
     1  'B22',12X,'G22',12X,'M22',/)                                       SHR 3/1
  211 FORMAT(I6,I3,1P7E15.7)                                               SHR 3/1
  212 IF(N .EQ. N1P  )WRITE(6,210)                                         SHR 3/1
      WRITE(6,211)NCYCLE,N,AMS(N),VS(N),QS(N),A22,BM22,G22,F22             SHR 3/1
C           OUTPUT OF MS(N2B)     4/5/76                                   SHR 3/1
      IF(N .EQ. N2B-1)WRITE(6,211)NCYCLE,N2B,AMS(N2B)                      SHR 3/1
  218 CONTINUE                                                             SHR 3/1
*IDENT  MSQSVS
*I  RPSL1D.5
C         PLOTTING MS, QS, AND VS.  3/19/76                                MSQSVS
*I  PDATA.52
      WRITE(NPLOT)(AMS(N),QS(N),VS(N),N=N1B,N2B)                           MSQSVS
*IDENT  BSTRS
*I  RPSL1D.5
C         NEW UNIAXIAL STRESS-STRAIN CURVE FOR BEAMS.  4/1/77              BSTRS
C         NOT COMPATIBLE WITH STREN OPTION.                                BSTRS
*D  MAIN.5
     3 SIG1(103,6),SIG2(103,6),LMAT(103,6),                                BSTRS
*D  MAIN.26
      COMMON SNRM(103),SNKM(103),SIG1M(103,6),SIG2M(103,6),LMATM(103,6)    BSTRS
*I  MAIN.28
      COMMON EPSR(103,6),EPSRM(103,6),EPLR(103,6),EPLRM(103,6),            BSTRS
     2  EFR(100),SSR(100),NEST,EPSZ                                        BSTRS
*I  RPSL1D.60
      EPSR (N,K) = 0.0                                                     BSTRS
      EPSRM(N,K) = 0.0                                                     BSTRS
      EPLR (N,K) = 0.0                                                     BSTRS
      EPLRM(N,K) = 0.0                                                     BSTRS
*D  START.16,START.20
      DO 700 I=1, 99                                                       BSTRS
      READ(5,702)EFR(I),SSR(I)                                             BSTRS
      IF(EFR(I) .LT. 0.0) GOTO 701                                         BSTRS
  700 CONTINUE                                                             BSTRS
  701 NEST = I                                                             BSTRS
      NSFL = 1                                                             BSTRS
      SSR(I) = SSR(I-1)                                                    BSTRS
      EFR(I) = 1000.                                                       BSTRS
      SSR(1) = SIGZ                                                        BSTRS
      EPSZ = SIGZ/E                                                        BSTRS
      EFR(1) = EPSZ                                                        BSTRS
      WRITE(6,703)(I,EFR(I),SSR(I),I=1,NEST)                               BSTRS
      IF(SSR(NEST) .GT. 2.0*SIGZ)WRITE(6,704)NEST                          BSTRS
  702 FORMAT(2F15.7)                                                       BSTRS
  703 FORMAT(/3X,'STRESS-STRAIN TABLE FOR BSTRS'/3X,'I',6X,'EPS(I)',9X,    BSTRS
     2 'SIG(I)',/(I5,1P2F15.7))                                            BSTRS
  704 FORMAT(//' ***WARNING*** STRAIN ENERGY SUSPICIOUS.   SIG(',I3,       BSTRS
     2 ') .GT. 2*SIGZ'//)                                                  BSTRS
*D  BMSTRS.23,BMSTRS.30
C         CHANGES IN BMSTRS FOR BSTRS 4/1/77                               BSTRS
      IF(IP .EQ. 1)GOTO 10                                                 BSTRS
      BEPS = EPSRM(N,K)                                                    BSTRS
      BEPL = EPLRM(N,K)                                                    BSTRS
      GOTO 11                                                              BSTRS
   10 BEPS = EPSR(N,K)                                                     BSTRS
      BEPL = EPLR(N,K)                                                     BSTRS
   11 BEPLM = BEPL                                                         BSTRS
      IF(BEPLM .GE. 0.0) GOTO 13                                           BSTRS
```

```
      SIG22I = -SIG22I                                                  BSTRS
      DEPS22 = -DEPS22                                                  BSTRS
      BEPS = -BEPS                                                      BSTRS
      BEPL = -BEPL                                                      BSTRS
   13 BEPS = BEPS + DEPS22                                              BSTRS
      IF(DEPS22 .GE. 0.0)GOTO 14                                        BSTRS
      IF(BEPS .GE. BEPL-EPSZ)GOTO 15                                    BSTRS
      IF(BEPS .GT. -EPSZ)GOTO 16                                        BSTRS
      CALL DVDINT( -BEPS,SIG22,EEB,SSB,NEST,2)                          BSTRS
      SIG22 = -SIG22                                                    BSTRS
      GOTO 17                                                           BSTRS
   14 IF(BEPS .LE. BEPL+EPSZ)GOTO 15                                    BSTRS
      CALL DVDINT(BEPS,SIG22,EEB,SSB,NEST,2)                            BSTRS
      BEPL = BEPS - EPSZ                                                BSTRS
      LMNK = 1                                                          BSTRS
      GOTO 18                                                           BSTRS
   15 SIG22 = SIG22I + E*DEPS22                                         BSTRS
      GOTO 18                                                           BSTRS
   16 CALL DVDINT(BEPS+2.0*EPSZ,SIG22,EEB,SSB,NEST,2)                   BSTRS
      SIG22 = SIG22 - 2.0*SIGZ                                          BSTRS
   17 LMNK = 1                                                          BSTRS
      BEPL = BEPS + EPSZ                                                BSTRS
   18 IF(BEPLM .GE. 0.0)GOTO 19                                         BSTRS
      SIG22 = -SIG22                                                    BSTRS
      BEPS = -BEPS                                                      BSTRS
      BEPL = -BEPL                                                      BSTRS
   19 IF(IP .EQ. 1) GOTO 20                                             BSTRS
      EPSHM(N,K) = BEPS                                                 BSTRS
      EPLBM(N,K) = BEPL                                                 BSTRS
      GOTO 21                                                           BSTRS
   20 EPSB(N,K) = BEPS                                                  BSTRS
      EPLB(N,K) = BEPL                                                  BSTRS
   21 CONTINUE                                                          BSTRS
*IDENT  APLFRC
*I  BPSLID.5
C         APPLIED FORCE (AND MOMENT) AS END CONDITION 4.   4/12/76      APL FRC
C         APL FRC ASSUMES SHR 3/1 IS INCLUDED IN THE DECK.              APL FRC
C         THE APPLIED FORCE CHANGE TO (1-D) BEPSIL FOR BEAMS REQUIRES   APL FRC
C         A USER SUBROUTINE, ENDFRC, TO SUPPLY END FORCES AND MOMENTS.  APL FRC
*I  MAIN.28
      COMMON EFR1,EFZ1,EM1,EFR2,EFZ2,EM2                                APL FRC
*I  START.54
      IF(IBCF1 .EQ. 4)N1V=N1B                                           APL FRC
*B  START.230
      IF(IBCF1 .LT. 5 .AND. IBCE2 .LT. 5)RETURN                         APL FRC
*D GRAD.9,GRAD.10
      IF(IBCF1 .NE. 4 .OR. N .GT. N1B)GOTO 8                            APL FRC
C        IBCF1=4 AND N=N1B.  USE FORWARD DIFFERENCES                    APL FRC
       R2 = RTD2*(-3.0* R(N)+4.0* R(N+1)- R(N+2))                       APL FRC
       Z2 = RTD2*(-3.0* Z(N)+4.0* Z(N+1)- Z(N+2))                       APL FRC
      DR2 = RTD2*(-3.0*DR(N)+4.0*DR(N+1)-DR(N+2))                       APL FRC
      DZ2 = RTD2*(-3.0*DZ(N)+4.0*DZ(N+1)-DZ(N+2))                       APL FRC
       R22 = RD22*(2.0* R(N)-5.0* R(N+1)+4.0* R(N+2)- R(N+3))           APL FRC
       Z22 = RD22*(2.0* Z(N)-5.0* Z(N+1)+4.0* Z(N+2)- Z(N+3))           APL FRC
      DR22 = RD22*(2.0*DR(N)-5.0*DR(N+1)+4.0*DR(N+2)-DR(N+3))           APL FRC
      DZ22 = RD22*(2.0*DZ(N)-5.0*DZ(N+1)+4.0*DZ(N+2)-DZ(N+3))           APL FRC
      GOTO 20                                                           APL FRC
    8 IF(IBCE2 .NE. 4 .OR. N .LT. N2B)GOTO 9                            APL FRC
C        IBCE2=4 AND N=N2B.  USE BACKWARD DIFFERENCES.                  APL FRC
*D  GRAD.34,GRAD.35
      IF(IBCF1 .NE. 4 .OR. N .GT. N1B  )GOTO 18                         APL FRC
C IBCF1=4 AND N=N1B  (+1/2).   USE NON-CENTRAL 4 POINT 2ND DIFFERENCES  APL FRC
       R22 = RD22M*(3.0* R(N)-7.0* R(N+1)+5.0* R(N+2)- R(N+3))          APL FRC
```

```
      Z22 = RD22M*(3.0* Z(N)-7.0* Z(N+1)+5.0* Z(N+2)- Z(N+3))          APL FRC
      DR22 = RD22M*(3.0*DR(N)-7.0*DR(N+1)+5.0*DR(N+2)-DR(N+3))         APL FRC
      DZ22 = RD22M*(3.0*DZ(N)-7.0*DZ(N+1)+5.0*DZ(N+2)-DZ(N+3))         APL FRC
      GOTO 20                                                          APL FRC
   18 IF(IBCE2 .NE. 4 .OR. N .LT. N2B-1)GOTO 19                        APL FRC
C IBCE2=4 AND N=N2B-1 (+1/2).  USE NON-CENTRAL 4 POINT 2ND DIFFERENCES  APL FRC
*D  RESULT.35,RESULT.49
      IF(N .EQ. N2B .AND. IBCE2 .EQ. 4)GOTO 13                         APL FRC
      IF(N .NE. N1B .OR. IBCE1 .NE. 4)GOTO 16                          APL FRC
C        N=N1B AND IBCE1=4.  APPLIED FORCE END1.                       APL FRC
      CALL ENDFRC                                                      APL FRC
      AMS(N1B) = EM1                                                   APL FRC
      GOTO 18                                                          APL FRC
C        N = N2B AND IBCE2=4.  APPLIED FORCE END2.                     APL FRC
   13 IF(IBCE1 .NE. 4)CALL ENDFRC                                      APL FRC
      AMS(N2B) = EM2                                                   APL FRC
      GOTO 18                                                          APL FRC
*D  MOTION.11,MOTION.18
C        APPLIED FORCE (AND MOMENT) AS END CONDITION 4.   4/12/76      APL FRC
      IF(IBCE1 .NE. 4 .OR. N .GT. N1B+1)GOTO 13                        APL FRC
      IF(N .GT. N1B)GOTO 11                                            APL FRC
C        N=N1B AND IBCE1=4.  USE DIFFERENCE (F(N+1/2)-F(N))/(DETA2/2).  APL FRC
      VR = 2.0*RTD2M*(VS(N)*SNRM(N)-QS(N)*SNKM(N)-EFR1)                APL FRC
      VZ = 2.0*RTD2M*(VS(N)*SNKM(N)+QS(N)*SNRM(N)-EFZ1)                APL FRC
      GOTO 35                                                          APL FRC
   11 M=N-1                                                            APL FRC
C        N=N1B+1 AND IBCE2=4.  USE DIFFERENCE (F(N+1/2)-F(N-1/2))/DETA2 APL FRC
      VR=RTD2M*(VS(N)*SNRM(N)-QS(N)*SNKM(N)-VS(M)*SNRM(M)+QS(M)*SNKM(M)) APLFRC
      VZ=RTD2M*(VS(N)*SNKM(N)+QS(N)*SNRM(N)-VS(M)*SNKM(M)-QS(M)*SNRM(M)) APLFRC
      GOTO 35                                                          APL FRC
   13 IF(IBCE2 .NE. 4 .OR. N .LT. N2B-1)GOTO 31                        APL FRC
      IF(N .LT. N2B)GOTO 16                                            APL FRC
C        N=N2B AND IBCE2=4.  USE DIFFERENCE (F(N)-F(N-1/2))/(DETA2/2)  APL FRC
      M=N2B-1                                                          APL FRC
      VR = 2.0*RTD2M*(EFR2-VS(M)*SNRM(M)+QS(M)*SNKM(M))                APL FRC
      VZ = 2.0*RTD2M*(EFZ2-VS(M)*SNKM(M)-QS(M)*SNRM(M))                APL FRC
      GOTO 35                                                          APL FRC
   16 M = N2B-2                                                        APL FRC
C        N=N2B-1 AND IBCE2=4.  USE DIFFERENCE (F(N+1/2)-F(N-1/2))/DETA2 APL FRC
      VR=RTD2M*(VS(N)*SNRM(N)-QS(N)*SNKM(N)-VS(M)*SNRM(M)+QS(M)*SNKM(M)) APLFRC
      VZ=RTD2M*(VS(N)*SNKM(N)+QS(N)*SNRM(N)-VS(M)*SNKM(M)-QS(M)*SNRM(M)) APLFRC
      GOTO 35                                                          APL FRC
*I  ROUNDR.9
      IF(IBCE1 .EQ. 4)GOTO 20                                          APL FRC
*I  ROUNDU.7
      IF(IBCE1 .EQ. 4)GOTO 20                                          APL FRC
*AF
*DECK  ENDFRC
      SUBROUTINE ENDFRC                                                 ENDFRC
C        THE APL FRC CHANGE TO (1-D) REPSIL FOR BEAMS REQUIRES A USER   ENDFRC
C        SUBROUTINE ENDFRC TO SUPPLY FORCES AND MOMENTS AT BOTH ENDS.   ENDFRC
C        FOR AN INITIALLY STRAIGHT BEAM. R(N)=0.0. Z(N)=(N-2)DETA2.     ENDFRC
C        EFZ1>0 DECREASES DZ(2).  EFR1>0 DECREASES DR(2)                ENDFRC
C        EFZ2>0 INCREASES DZ(N2B). EFR2>0 INCREASES DR(N2B)             ENDFRC
C        EM1>0 DECREASES DR(2)      AND INCREASES DR(3)                 ENDFRC
C        EM2>0 INCREASES DR(N2B-1) AND DECREASES DR(N2B)                ENDFRC
C                                                                       ENDFRC
*CALL MAIN
C        TABULAR INPUT FOR EFZ1, EFTH, AND EFMU.  10/27/76             10/27/76
C        MODIFIED  (12/7/76)  FORCE RELATIVE TO NORMAL                  12/7/76
C      EFR1 = EFMU*EFZ1.  EM1 = ZU*(EFZ1*SIN(EFTH)+EFR1*COS(EFTH))      10/27/76
      DIMENSION LABELV(7),EFZT(40),TMFZ(40),EFTHT(40),TMTH(40),         3/9/77
     1 EFMUT(40),TMMU(40)                                               3/8/77
```

```
      DATA IENDFR/0/                                                    10/27/76
      IF(IENDFR .GT. 0)GOTO 1                                           10/27/76
      IENDFR = 1                                                        10/27/76
      READ (5,100)NFZPTS,(LABELV(I),I=1,7)                              10/27/76
      WRITE(6,101) (LABELV(I),I=1,7),NFZPTS                             10/29/76
      READ (5,102)(  EFZT (I),TMFZ(I),I=1,NFZPTS)                       10/27/76
      WRITE(6,103)(I,EFZT (I),TMFZ(I),I=1,NFZPTS)                       10/27/76
      READ (5,100)NFTHS ,(LABELV(I),I=1,7)                              10/27/76
      WRITE(6,104) (LABELV(I),I=1,7),NFTHS                              10/29/76
      READ (5,102)(  EFTHT(I),TMTH(I),I=1,NFTHS)                        10/27/76
      WRITE(6,103)(I,EFTHT(I),TMTH(I),I=1,NFTHS)                        10/27/76
      READ (5,100)NFMUS ,(LABELV(I),I=1,7)                              10/27/76
      WRITE(6,105) (LABELV(I),I=1,7),NFMUS                              10/29/76
      READ (5,102)(  EFMUT(I),TMMU(I),I=1,NFMUS)                        10/27/76
      WRITE(6,103)(I,EFMUT(I),TMMU(I),I=1,NFMUS)                        10/27/76
      EFR2 = 0.0                                                        10/27/76
      EFZ2 = 0.0                                                        10/27/76
      EM2  = 0.0                                                        10/27/76
    1 CALL DVDINT(TIME,EFZ1,TMFZ,EFZT,NFZPTS,2)                         10/27/76
      CALL DVDINT(TIME,EFTH,TMTH,EFTHT,NFTHS,2)                         10/27/76
      CALL DVDINT(TIME,EFMU,TMMU,EFMUT,NFMUS,2)                         10/27/76
      EFR1 = EFMU*EFZ1                                                  10/27/76
      EM1= -7U*(SIN(EFTH)*EFZ1 + (1.0-COS(EFTH))*EFR1)                  10/29/76
C        CODE CHECKING PLOTS.  10/28/76                                 CC/10/76
      P(53) = EFZ1                                                       7/20
C        REPLACE COMPUTATION OF EFZ1 AND EFZ2 IN ENDFRC 12/7/76 CARDS    EF0D21
C        INTERPOLATE FOR NORMAL AT END WITH QUADRATIC INTERPOLATION       EF0D21
      XMX1 = DE - DETA2                                                 EF0D21
      XMX2 = DE - 2.0*DETA2                                             EF0D21
C        X-X0 = DE.    X2-X1 = X1-X0 = DETA2.                          EF0D21
      SNREND=(XMX2*(XMX1*SNR(N1B)-2.0*DF*SNR(N1B+1))+DE*XMX1*SNR(N1B+2) EF0D21
     1  )*RD22M                                                         EF0D21
      SNKEND=(XMX2*(XMX1*SNK(N1B)-2.0*DF*SNK(N1B+1))+DE*XMX1*SNK(N1B+2) EF0D21
     1  )*RD22M                                                         EF0D21
      AAA = SQRT(SNREND**2 + SNKEND**2)                                 EF0D21
      SNREND = SNREND/AAA                                               EF0D21
      SNKEND = SNKEND/AAA                                               EF0D21
      TEFR1 = EFR1*SNREND - EFZ1*SNKEND                                 EF0D21
      EFZ1  =EFR1*SNKEND + EFZ1*SNREND                                  EF0D21
      EFR1 = TEFR1                                                      EF0D21
C     TEFR1 = EFR1*SNR(N1H) - EFZ1*SNK(N1H)                             12/7/76
C     EFZ1 = EFR1*SNK(N1B) + EFZ1*SNR(N1H)                              12/7/76
C     EFR1 = TEFR1                                                      12/7/76
      RETURN                                                            10/27/76
  100 FORMAT(I10,7A10)                                                  10/27/76
  101 FORMAT('1SUBROUTINE ENDFRC 10/27/76. TABULAR DATA EFZ1,EFTH,EFMU',10/27/76
     1       /' TABLE 1.',7A10,I5,' POINTS',/4X,'I',6X,'EFZT ',9X,'TIME')10/27/76
  102 FORMAT(2E15.7)                                                    10/27/76
  103 FORMAT(I5,1P2E15.7)                                               10/27/76
  104 FORMAT(/' TABLE 2.',7A10,I5,' POINTS',/4X,'I',6X,'EFTHT',9X,'TIME')10/27/76
  105 FORMAT(/' TABLE 3.',7A10,I5,' POINTS',/4X,'I',6X,'EFMUT',9X,'TIME')10/27/76
      END                                                               10/27/76
*IDENT  EAPFRC
*I  APSL1D,5
C        EXTERNAL WORK FOR APL FRC OF 4/12/76.   (6/3/76)               EAP FRC
*I  MAIN,2A
      COMMON DWF1P,DTH1P,DWF2P,DTH2P                                    EAP FRC
*I  APSL1D,43
      DTH1P = 0.0                                                       EAP FRC
      DWF1P = 0.0                                                       EAP FRC
      DWF2P = 0.0                                                       EAP FRC
      DTH2P = 0.0                                                       EAP FRC
*D  MOTION,40
```

```
      IF(N .GT. N1B)GOTO 130                                                    EAP FRC
      DRI1 = DRI                                                                EAP FRC
      DZI1 = DZI                                                                EAP FRC
  130 CONTINUE                                                                  EAP FRC
      IF(IBCE1 .NE. 4)GOTO 131                                                  EAP FRC
      N=N1B                                                                     EAP FRC
      DWF =-0.5*(EFR1*(DRI1+DR(N))+EFZ1*(DZI1+DZ(N)))                           EAP FRC
      R2=RTD2*(-3.0*(R(N)+DR(N))+4.0*(R(N+1)+DR(N+1))-(R(N+2)+DR(N+2)))         EAP FRC
      Z2=RTD2*(-3.0*(Z(N)+DZ(N))+4.0*(Z(N+1)+DZ(N+1))-(Z(N+2)+DZ(N+2)))         EAP FRC
      D = SQRT(Z2**2 + R2**2)                                                   EAP FRC
      SNRN =  Z2/D                                                              EAP FRC
      SNKN = -R2/D                                                              EAP FRC
      DTHET = ARCSIN(SNK(N)*SNRN - SNR(N)*SNKN)                                 EAP FRC
      DWM =-0.5*EM1*(DTH1P + DTHET)                                             EAP FRC
      TNRG = TNRG + 0.5*(DWF + DWM + DWE1P)                                     EAP FRC
      DWE1P = DWF + DWM                                                         EAP FRC
      DTH1P = DTHET                                                             EAP FRC
  131 IF(IBCE2 .NE. 4)GOTO 132                                                  EAP FRC
      N=N2B                                                                     EAP FRC
      DWF = 0.5*(EFR2*(DRI+DRS) + EFZ2*(DZI+DZS))                               EAP FRC
      R2=RTD2*(R(N-2)+DR(N-2)-4.0*(R(N-1)+DR(N-1))+3.0*(R(N)+DR(N)))            EAP FRC
      Z2=RTD2*(Z(N-2)+DZ(N-2)-4.0*(Z(N-1)+DZ(N-1))+3.0*(Z(N)+DZ(N)))            EAP FRC
      D = SQRT(Z2**2 + R2**2)                                                   EAP FRC
      SNRN =  Z2/D                                                              EAP FRC
      SNKN = -R2/D                                                              EAP FRC
      DTHET = ARCSIN(SNK(N)*SNRN - SNR(N)*SNKN)                                 EAP FRC
      DWM =-0.5*EM2*(DTH2P + DTHET)                                             EAP FRC
      TNRG = TNRG + 0.5*(DWF + DWM + DWE2P)                                     EAP FRC
      DWE2P = DWF + DWM                                                         EAP FRC
      DTH2P = DTHET                                                             EAP FRC
  132 CONTINUE                                                                  EAP FRC
*IDENT  11/2/76
*I  RPSL1D.5
C         INSERT TO REMOVE FAILURE HALT AT FREE OR FORCED END.  11/2/76   11/2/76
*I  RMSTRS.13
C         INSERT TO REMOVE FAILURE HALT AT FREE OR FORCED END.  11/2/76   11/2/76
      IF(G22 .GT. 0.0)GOTO 2                                                    11/2/76
      IF(IP .EQ. 2)GOTO 2                                                       11/2/76
      IF(N .EQ. N1B .AND. IBCE1.EQ.4)G22=1.0E-10                                11/2/76
      IF(N .EQ. N2B .AND. IBCE2.EQ.4)G22=1.0E-10                                11/2/76
    2 CONTINUE                                                                  11/2/76
*IDENT  INVELC
*I  INVEL.6
C         LATERAL VELOCITY . VR. TOWARD END 1.                                  1/8/76
C              ENTIRE ROD MOVING WITH CONSTANT VELOCITY.  7/20/76
      WRITE(6,333)                                                              1/8/76
  333 FORMAT(///' LATERAL VELOCITY TOWARD END 1.  1/8/76'//)                    1/8/76
      DO 29 N=N1V,N2V                                                           7/20/76
      DZ(N)=-VR                                                                 1/8/76
   29 WRITE(6,666)N,DZ(N),DR(N)                                                 1/8/76
      RETURN                                                                    1/8/76
  666 FORMAT(' N=',I3,'  DZ(N)=  ',1PE13.6,'  DR(N)=  ',E13.6)                  1/8/76
*IDENT  EROD26
*I  RPSL1D.5
C         EROSION OF END1.  12/26/76.  PUT AFTER APLFRC AND SHR3/1.             EROD
C         INCLUDE SUBROUTINE ERODE AND CORRECT SUBROUTINE ENDFRC.
*I  MAIN.28
      COMMON DE                                                                 EROD21
*I  RPSL1D.129
      CALL ERODE                                                                EROD21
*I  RESULT.61
      IF(N .EQ. N1B)VS(N)=(AMS(N+1)-AMS(N))/(SRA*(DETA2-DE))
      IF(N .EQ. N1B)GOTO 209
```

```
*I  MOTION.26
      IF(N .NE. N1B .OR. IBCE1 .NE. 4)GOTO36
C        REPLACES VR,VZ COMPUTATION IN MOTION WHEN N=N1B AND IBCE1=4      ERODE21
      VR = (VS(N)*SNRM(N)-QS(N)*SNRM(N)-FFR1)/(0.5*DETA2 -DF)             ERODE21
      VZ = (VS(N)*SNRM(N)+QS(N)*SNRM(N)-FFZ1)/(0.5*DETA2 - DF)            ERODE21
   36 CONTINUE
*I  STRAIN.81
      IF(J1 .GE. N1B) GOTO 44                                             ERODE21
      EPSS1(I) = EPSS2(I) = 0.0                                           ERODE21
      EPSANG(I) = EPSANP(I) = -1.0E20                                     ERODE21
      GOTO 46                                                             ERODE21
   44 CONTINUE                                                            ERODE21
*I  STRAIN.97
      IF(NQ1 .GE. N1H)GOTO 49                                             ERODE21
      D1 = D2 = 0.0                                                       ERODE21
      GOTO(71,50),LINK                                                    ERODE21
   49 CONTINUE                                                            ERODE21
*I  PDATA.7
      N1B = 2                                                             ERODE21
*D  PDATA.61
  100 N1B = N1A
      RETURN                                                              ERODE21
*AF
*DECK  ERODE
      SUBROUTINE ERODE                                                    ERODE21
C        INSERT ANY INPUT FOR ERODE AFTER INPUT FOR ENDFRC.                ERODE21
C        THIS USER  SUBROUTINE COMPUTES THE EROSION ON END1 OF A ROD.      ERODE21
C        THIS VERSION ASSUMES TABLE OF EROSION FROM END OF ROD VS TIME.    ERODE26
C        THE TABLE PRESUMABLY AGREES WITH ENDFRC VS TIME TABLE IN ENDFRC   ERODE26
C        AND STRESS-STRAIN TABLE READ IN INPUT.                            ERODE26
*CALL  MAIN
      DIMENSION LABLTE(7),TEROD(100),XEND(100)                            ERODE26
      DATA IERODE/0/                                                      ERODE21
      IF(IERODE .EQ. 1)GOTO 10                                            ERODE21
      DECOM = DETA2/4.0
      IERODE = 1                                                          ERODE21
      EROOP = 0.0                                                         ERODE26
      DF = 0.0
      READ(5,100)NTEPTS,(LABLTE(I),I=1,7)                                 ERODE26
      WRITE(6,101)NTEPTS,(LABLTE(I),I=1,7)                                ERODE26
      READ (5,102)(  TEROD(I),XEND(I),I=1,NTEPTS)                         ERODE26
      WRITE(6,103)(I,TEROD(I),XEND(I),I=1,NTEPTS)                         ERODE26
  100 FORMAT(I10,7A10)                                                    ERODE26
  101 FORMAT('1SUBROUTINE ERODE 5/26/77, TABULAR DATA TIME VS XEND'/      ERODE26
     2  I5,' POINTS',5X,7A10,//4X,'I',7X,'TIME',9X,'XEND')                ERODE26
  102 FORMAT(2E15.7)                                                      ERODE26
  103 FORMAT(I5,2E15.7)                                                   ERODE26
   10 CALL DVDINT(TIME,EROD,TEROD,XEND,NTEPTS,2)                          ERODE26
      DELDF = EROD - EROOP                                                ERODE26
      DF = DF + DELDF                                                     ERODE26
      EROOP = EROD                                                        ERODE26
      P(54) = EROD                                                        ERODE26
C        CODE CHECKING PLOT 5/27/77                                        CC
   19 IF(DF .LT. DFCOM)GOTO 20                                            CC
      DF = DF -DETA2                                                      ERODE21
      P(N1B) = Z(N1B) = -1.0E20                                           ERODE21
C        THE FOLLOWING CARD IS NEEDED IF THE DECK INCLUDES   MSQSVS.      ERODE21
      AMS(N1B) = QS(N1B) = VS(N1B) = 0.0                                  ERODE21
      N1B = N1V = N1A = N1B+1                                             ERODE21
      GOTO 19                                                             ERODE21
   20 RETURN                                                              ERODE21
      END                                                                 ERODE21
?
```

```
        FRODING ROD  (254 BY 8.128 MM)     REPEAT OF 6/8/77 RUN
       36     6 2.0
 130    0 9999  .5000000E-6
        4        4
   0    0 9999 68.94757      .001
1.985714E11 .27          1.3900000E9 7790.        .008128          4     0
0.007            1.390        E9
0.008            1.518        E9
0.009            1.564        E9
0.010            1.597        E9
0.011            1.624        E9
0.012            1.648        E9
0.013            1.669        E9
0.014            1.688        E9
0.015            1.705        E9
0.016            1.721        E9
0.017            1.736        E9
0.018            1.750        E9
0.019            1.763        E9
0.020            1.776        E9
0.022            1.799        E9
0.024            1.820        E9
0.026            1.840        E9
0.028            1.860        E9
0.030            1.879        E9
0.032            1.897        E9
0.034            1.915        E9
0.036            1.932        E9
0.038            1.949        E9
0.040            1.966        E9
0.045            2.008        E9
0.050            2.048        E9
0.055            2.086        E9
0.060            2.126        E9
0.065            2.164        E9
0.070            2.202        E9
0.075            2.239        E9
0.080            2.276        E9
0.085            2.312        E9
0.090            2.348        E9
0.095            2.384        E9
0.100            2.419        E9
0.105            2.454        E9
0.110            2.489        E9
0.115            2.523        E9
0.120            2.557        E9
0.125            2.590        E9
0.130            2.624        E9
0.135            2.656        E9
0.140            2.689        E9
0.145            2.721        E9
0.150            2.753        E9
0.30             3.713        E9
-1.0
  20    1    1    1
   2   70  130
   2   70  130
   2   70  130
   2   53   54
          0.021167     6
          0.005      45.0       135.0         0
          0.010      45.0       135.0         0
          0.015      45.0       135.0         0
```

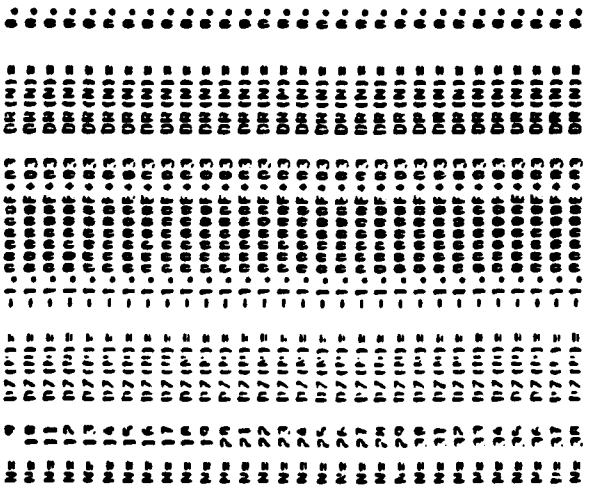
```
          .01926      45.0        135.0          0
          .03967      45.0        135.0          0
            .06       45.0        135.0          0
 .254       .004064       1
 .004064
4.19606E-31.35467E-3-3.3543E-3
6.98023E-31.35467E-3-2.0126E-3
7.97487E-31.35467E-3-6.7086E-4
7.97487E-31.35467E-3 6.7086E-4
6.98023E-31.35467E-3 2.0126E-3
4.19606E-31.35467E-3 3.3543E-3
                     1000.
        29  EFZ1 AND TIME     6/3/77  CASE B
 0.0                 0.0
  72160.00           1.080      E-6
  78750.00           1.590      E-6
  81140.00           1.790      E-6
  82860.00           2.143      E-6
  85550.00           2.715      E-6
  87650.00           2.915      E-6
  89380.00           3.115      E-6
  90890.00           3.315      E-6
  92220.00           3.515      E-6
  95090.00           3.890      E-6
  97600.00           4.372      E-6
  99940.00           4.847      E-6
 102160.00           5.257      E-6
 104310.00           5.579      E-6
 106390.00           5.968      E-6
 110430.00           6.580      E-6
 114350.00           7.123      E-6
 118200.00           7.574      E-6
 121960.00           7.943      E-6
 125650.00           8.240      E-6
 129260.00           8.495      E-6
 132790.00           8.717      E-6
 136250.00           8.917      E-6
 139630.00           9.117      E-6
 142940.00           9.317      E-6
 159310.00           9.817      E-6
 192050.00          19.817      E-6
 192050.00           1.00000
         2  THETA = 0.0.    3/8/77
 0.0                 0.0
 0.0                 1.0
         2 MU = 0.0.     3/8/77
 0.0                 0.0
 0.0                 1.0
        29     TIME AND XEND  6/3/77  CASE B
  0.000      E-6   3.000      E-3
  1.080      E-6   6.000      E-3
  1.590      E-6   9.000      E-3
  1.790      E-6   9.256      E-3
  2.143      E-6   9.256      E-3
  2.715      E-6   9.256      E-3
  2.915      E-6   9.403      E-3
  3.115      E-6   9.482      E-3
  3.315      E-6   9.644      E-3
  3.515      E-6   9.855      E-3
  3.890      E-6   9.855      E-3
  4.372      E-6   9.855      E-3
  4.847      E-6   9.855      E-3
  5.257      E-6   9.855      E-3
```

```
  5.579      E-6  9.855      E-3
  5.968      E-6  9.855      E-3
  6.580      E-6  9.855      E-3
  7.123      E-6  9.855      E-3
  7.574      E-6  9.855      E-3
  7.943      E-6  9.855      E-3
  8.240      E-6  9.855      E-3
  8.495      E-6  9.855      E-3
  8.717      E-6  9.855      E-3
  8.917      E-6  9.864      E-3
  9.117      E-6  9.893      E-3
  9.317      E-6  9.915      E-3
  9.817      E-6  9.915      E-3
 19.817      E-6  9.915      E-3
  1.00000         9.915      E-3
?
*IDENT PLOTP
*D  RP1PLT.18
      READ(NPLOT)NNPE,(PPF(I),I=1,NNPF)                                    PLOTP
*D  RP1PLT.40
     1 ,(PDAT(J),J=1,NNPF)                                                 PLOTP
*ID  MSQSVS
*D  MAIN.10
C          PLOTTING MS, QS, AND VS.  3/19/76                              MSQSVS
      COMMON/MSQSVS/ AMS(103),QS(103),VS(103),XSN(103)                     MSQSVS
      LEVEL2,TIM,PC,AMS                                                    MSQSVS
*I  RP1PLT.61
C          PLOTTING MS, QS, AND VS.  3/19/76                              MSQSVS
      READ (NPLOT)(AMS(N),QS(N),VS(N),N=1,N1)                              MSQSVS
      IF(EOF(NPLOT) .NE. 0)GOTO 28                                         MSQSVS
*D  PLOT3D.53
      XMAR = 3.0                                                           MSQSVS
*I  PLOT3D.93
C          PLOTTING MS, QS, AND VS.  3/19/76                              MSQSVS
      IF(NCYCLE .EQ. 0) RETURN                                             MSQSVS
      ENCODE(10,711,HDR)NCYCLE                                             MSQSVS
  711 FORMAT(5HCYCLE,I4,1H>)                                               MSQSVS
      HDR1 = 10HMESH PTS.>                                                 MSQSVS
      CALL PLTPGE                                                          MSQSVS
      DO 72 I=1,I2                                                         MSQSVS
      XSN(I) = FLOAT(I)+1.0                                                MSQSVS
   72 CONTINUE                                                             MSQSVS
      CALL FIXSCA(XSN(1),  I2,7.9,DSXS,XMINS,XMAXS,DELDXS)                 MSQSVS
      CALL FIXSCA(AMS(1),  I2,6.4,DSYS,YMINS,YMAXS,DELDYS)                 MSQSVS
      CALL PLTSCA(3.0, 2.0,0.0,0.0,1.0,1.0)                                MSQSVS
      HD = 5H   M>                                                         ANYUNITS
      CALL PLTSYM(.1,HD, 90.0,  -1.2,2.5)                                  MSQSVS
      CALL PLTSYM(.1,HDR,0.0,    3.5,-1.0)                                 MSQSVS
      CALL PLTSYM(.1,HDR1,0.0,   3.5,-0.6)                                 MSQSVS
      CALL PLTSCA(3.0, 2.0,XMINS,YMINS,DSXS,DSYS)                          MSQSVS
      CALL PLTPT2(1.0,XSN(1),AMS(1),  I2,0)                                MSQSVS
      CALL PLTAXS(DELDXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,4)                 MSQSVS
      CALL LABELA(DELDXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,1.0,1.0)           MSQSVS
      I2M1 = I2-1                                                          MSQSVS
      DO 73 I=1,I2M1                                                       MSQSVS
      XSN(I) = XSN(I)+0.5                                                  MSQSVS
   73 CONTINUE                                                             MSQSVS
      CALL FIXSCA( QS(1),I2M1,6.4,DSYS,YMINS,YMAXS,DELDYS)                 MSQSVS
      CALL PLTSCA(3.0,12.0,0.0,0.0,1.0,1.0)                                MSQSVS
      HD = 5H   Q>                                                         ANYUNITS
      CALL PLTSYM(.1,HD, 90.0,  -1.2,2.5)                                  MSQSVS
      CALL PLTSYM(.1,HDR,0.0,    3.5,-1.0)                                 MSQSVS
      CALL PLTSYM(.1,HDR1,0.0,   3.5,-0.6)                                 MSQSVS
```

```
      CALL PLTSCA(3.0,12.0,XMINS,YMINS,DSXS,DSYS)                          MSQSVS
      CALL PLTDT2(1.0,XSN(1), QS(1),I2M1,0)                                MSQSVS
      CALL PLTAXS(DFLDXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,4)                 MSQSVS
      CALL LABELA(DELDXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,1.0,1.0)           MSQSVS
      CALL FIXSCA( VS(1),I2M1,6,4,DSYS,YMINS,YMAXS,DFLDYS)                 MSQSVS
      CALL PLTSCA(3.0,22.0,0.0,0.0,0.0,1.0,1.0)                            MSQSVS
      HD = 5H    V>                                                        ANYUNITS
      CALL PLTSYM(,1,HD, 90.0,  -1.2,2.5)                                  MSQSVS
      CALL PLTSYM(,1,HDP,0.0,    3.5,-1.0)                                 MSQSVS
      CALL PLTSYM(,1,HDR1,0.0,    3.5,-0.6)                                MSQSVS
      CALL PLTSCA(3.0,22.0,XMINS,YMINS,DSXS,DSYS)                          MSQSVS
      CALL PLTDT2(1.0,XSN(1), VS(1),I2M1,0)                                MSQSVS
      CALL PLTAXS(DFLDXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,4)                 MSQSVS
      CALL LABELA(DELDXS,DELDYS,XMINS,XMAXS,YMINS,YMAXS,1.0,1.0)           MSQSVS
      I = 4                                                                MSQSVS
?
 1.0        3.0
?
```

```
                                    HRL PEPSIL CODE

                    ERODING PND  (254 BY 8.128 MM)    REPEAT OF 6/8/77 RUN

                          COMPUTATIONS FOR ENERGY USE DETAIL= .1000000E+01
                       36 MESHES IN RIAP DIRECTION   (DETAIL= .7055566E-02)

                             PENDING  TIME INCREMENT=  .9426184E-05
                             PERFORMANCE TIME INCREMENT=  .1397478E-05
                             INPUT    TIME INCREMENT=  .5000000E-06

                           TIME INCREMENT USED BY REPSIL=  .5000000E-06

                    YOUNG'S MODULUS = .1985718+12
                    POISSON'S RATIO = .2700000E+00          YIELD STRESS    = .1390000E+10
                    MASS DENSITY    = .7790000E+04          THICKNESS       = .8128000E-02

                                    START AT TIME STEP    0
                                    FINAL  TIME  STEP   130
                                    SURFACE STRAINS EVERY   20 TIME STEP
                                    RESTART  WRITE  EVERY 9999 TIME STEP

                               LAYER =    6                  NSTEN  =    6
                               LOAD  =    0                  LPRESS =    0

                                    BOUNDARY CONDITIONS
            1/2/3/4/     = CLAMPED/SYMMETRY/HINGED/FREE/
                                 END1 (IRCE4) =   4
                                 END2 (IPCE7) =   4

                                 PRINT OPTION CONTROL CARD
                                   0/1 = NO PRINT/PRINT
                                    1 DISPLACEMENT INCREMENTS
                                    1 CARTESIAN COORDINATES, PRESSURE
                                    1 SURFACE NORMAL VECTOR COMPONENTS

            PRINT INFORMATION AT THE FOLLOWING TIME STEPS
               70  130
            PRINT L MATRIX (LPAT) AT THE FOLLOWING TIME STEPS
               70  130
            3-D PLOTS FOR THE FOLLOWING TIME STEPS
               70  130
                       CONSTITUTIVE RELATION    ELASTOPLASTIC-NO WORK HARDENING-STRAIN RATE INDEPENDENT

                                 STRESS-STRAIN AND STRAIN RATE PARAMETERS
                     J        SSIG(J)         SEPS(J)          DSR(J)          1/DSR(J)
                     1     1.3900000E+09  7.0000010E-03       0.              0.


                                 START DAMPING AFTER TIME STEP 9999     TIME = .5000E-02
                              DAMPP = .4995E+02                 DFACT = .1000E-02


LATERAL VELOCITY INBOARD END 1.  1/8/76


N=  2  VZ(N)=  -1.0000000E+03  DR(N)=   0.
N=  3  VZ(N)=  -1.0000000E+03  DR(N)=   0.
N=  4  VZ(N)=  -1.0000000E+03  DR(N)=   0.
N=  5  VZ(N)=  -1.0000000E+03  DR(N)=   0.
N=  6  VZ(N)=  -1.0000000E+03  DR(N)=   0.
N=  7  VZ(N)=  -1.0000000E+03  DR(N)=   0.
N=  8  VZ(N)=  -1.0000000E+03  DR(N)=   0.
```

Selected output from Example 2



TIME STEP 130 TIME .65000000E-04

DISPLACEMENT INCREMENTS BETWEEN T.S. 129 AND 130 — COORDINATES — PRESSURE

N	DR(N)	DZ(N)	R(N)	Z(N)	P(N)
1	0.	0.	0.	0.	0.
2	0.	-.5058571387301231E-03	-.1000000000000000E+21	-.1000000000000000E+21	0.
3	-.1106274105656040E-21	-.5304025600115780E-03	-.1000000000000000E+21	-.1000000000000000E+21	0.
4	.3039159285477446E-18	-.6741387301888412E-03	.1419663827520569E-16	-.6876392633523443E-01	0.
5	.2244574532291786E-18	-.6744651948159715E-03	.8029794325849791E-17	-.5910820402037508E-01	0.
6	.1571558027667517E-18	-.6741438873046832E-03	.3651180536943825E-17	-.4947479598178539E-01	0.
7	.6158047900898874E-19	-.6736144145305621E-03	.4087539516351190E-18	-.3985784426284670E-01	0.
8	.5318263691201667E-20	-.6680867618907919E-03	-.8324286468355139E-18	-.3026601321769210E-01	0.
9	-.2690865374360874E-19	-.6318120914938359E-03	-.1057718485732633E-17	-.2106659241692221E-01	0.
10	.1364531667380086E-19	-.5919287836303640E-03	-.3250018723704805E-18	-.1259741376366047E-01	0.
11	-.3917478315990637E-19	-.5633224574144308E-03	.1583777558484437E-17	-.4706753916223966E-02	0.
12	.8220180427577081E-19	-.5456871833099085E-03	-.2965793121458121E-18	.2825182306171653E-02	0.
13	-.7532610693966540E-19	-.5363426897401144E-03	-.5496783729650861E-18	.1016595267691961E-01	0.
14	.3278581115844279E-19	-.5315651702970693E-03	-.5605314370473627E-18	.1741537131607083E-01	0.
15	.6319171512572368E-19	-.5288488862263303E-03	.4831094848604567E-18	.2462052227469863E-01	0.
16	-.1014345975667750E-18	-.5269372883919963E-03	-.1326074986974812E-18	.3180033854861979E-01	0.
17	.6961402846738053E-19	-.5255157709517719E-03	.9522439080353268E-19	.3896319533001913E-01	0.
18	-.8527241873462777E-19	-.5246003718658484E-03	.5492683640912605E-18	.4611416265081036E-01	0.
19	.7367270292155854E-19	-.5234654212675590E-03	.1465630157859634E-18	.5312561712728312E-01	0.
20	.7791911123724712E-19	-.5226865164096691E-03	.1519639595110764E-17	.6039107877685113E-01	0.
21	-.6364887592270879E-19	-.5220961714129564E-03	-.2528929385805899E-18	.6752060270145632E-01	0.
22	-.3495361681966667E-20	-.5215867014876713E-03	-.1310523377777245E-17	.7464614397644542E-01	0.
23	-.1085315665524413E-18	-.5212360828357118E-03	-.1090137580165164E-17	.8176849140524123E-01	0.
24	.8541473043307835E-19	-.5210365614186328E-03	.4923364518887872E-18	.8888091911048136E-01	0.
25	-.4594831820111822E-19	-.5210940229921912E-03	.1919167880587874E-18	.9608710748853380E-01	0.
26	.5547706239995430E-19	-.5220924300694714E-03	-.1037289382967882E-17	.1031195277042589E+00	0.
27	-.7490824623213511E-19	-.5239814183537889E-03	.1389456037752231E-18	.1102253708304234E+00	0.
28	.3146311325491146E-19	-.5271847563374198E-03	.1394967281979882E-17	.1173721212174038E+00	0.
29	.9916735115625944E-19	-.5314111746015605E-03	.3125511465061690E-18	.1244072277145412E+00	0.
30	-.3277115213360882E-19	-.5356147620496216E-03	-.3329651514935745E-19	.1314797179548464E+00	0.
31	.1108512781179974E-19	-.5382571452848851E-03	-.2694176398454509E-18	.1385426373927379E+00	0.
32	-.4512561727768454E-19	-.5386682704848848E-03	-.9452328624924633E-20	.1456021051394254E+00	0.
33	-.3438862194877245E-19	-.5378790017058983E-03	-.4959559667491182E-18	.1526625414971461E+00	0.
34	-.2412749928722928E-19	-.5380389903004608E-03	-.3174671647149937E-18	.1597229373436280E+00	0.
35	.1532682931729844E-19	-.5386884035241340E-03	.4066648840750564E-19	.1667806735055883E+00	0.
36	.1486485433888125E-19	-.5389085971767940E-03	.2398694543165774E-18	.1738368572963171E+00	0.
37	.1721568838537241E-20	-.5389146121280401E-03	.1831583476571852E-18	.1808929086487840E+00	0.
38	.1194781804575857E-19	-.5393854479489352E-03	.3529713658218186E-18	.1879480685766082E+00	0.

CYCLE	N	PS	VS	QS	A22	B22	Q22	M22
130	4	0.	-1.9610239E-12	1.9997692E+05	1.8728677E+00	2.2510668E-14	1.0197032E+05	-2.4632903E-13
130	5	-3.8196270E-14	1.1229016E-12	1.9188456E+05	1.8642721E+00	1.5654638E-14	1.0291941E+05	2.1406882E-13
130	6	-1.9378515E-14	-1.5670397E-12	1.9265645E+05	1.8578577E+00	1.6892495E-14	1.0369817E+05	-2.4183921E-13
130	7	-3.4486354E-14	1.7899574E-12	1.9265645E+05	1.8481644E+00	1.6110131E-14	1.0424205E+05	2.1358574E-13
130	8	-1.7287385E-14	-3.8755101E-11	1.8080228E+05	1.7000376E+00	1.1491892E-14	1.0635193E+05	-1.0092683E-14
130	9	-3.7381187E-13	4.1518520E-11	1.5348768E+05	1.4408546E+00	1.5666167E-14	1.0652545E+05	-1.3781157E-14
130	10	-2.2251057E-14	-2.3589093E-11	1.2972235E+05	1.2507319E+00	-1.9167900E-14	1.0371716E+05	1.6412764E-14
130	11	-2.8775411E-13	2.4179345E-11	1.1238567E+05	1.1395958E+00	-2.0764732E-14	9.8618889E+04	2.4307697E-13
130	12	-2.5636825E-14	2.9885542E-12	1.0183460E+05	1.0824824E+00	1.7256167E-14	9.4075063E+04	-1.3405671E-14
130	13	-3.6085346E-15	-1.6684442E-12	9.6039655E+04	1.0557883E+00	1.2335267E-14	9.0971774E+04	-9.2666958E-15
130	14	-1.5793785E-14	5.5587298E-12	9.2731647E+04	1.0428545E+00	-5.7284641E-15	8.8970980E+04	-2.2316726E-13
130	15	2.6257782E-14	-5.0662695E-12	9.0483949E+04	1.0355337E+00	-7.9480581E-15	8.7301794E+04	2.3310367E-13
130	16	-1.2117182E-14	-8.8411574E-11	8.8540564E+04	1.0306474E+00	1.0434889E-14	8.5987224E+04	1.0628476E-13
130	17	-5.8809377E-13	6.7882544E-11	8.7020193E+04	1.0277287E+00	-6.1527645E-15	8.4713551E+04	1.1799069E-13
130	18	-1.8467819E-13	-5.8889170E-12	8.5698921E+04	1.0246565E+00	8.9997833E-15	8.3636730E+04	5.6221036E-13
130	19	-1.4887244E-13	4.2837485E-11	8.4487486E+04	1.0276280E+00	-1.3430639E-14	8.2610856E+04	3.5022366E-13

130 20	1.5900113E-13	8.8831799E-17	8.3458150E+04	1.0210773E+00	-2.3933015E-14	8.1735395E+04	-9.7532[illegible]E-14
130 21	2.2239397E-13	-4.9876513E-11	8.2633782E+04	1.0195369E+00	1.9615149E-14	8.1018527E+04	1.0856346E-13
130 22	-1.3380311E-13	8.4915179E-11	8.1866348E+04	1.0190227E+00	2.6823087E-14	8.0338089E+04	3.2379549E-13
130 23	3.2934529E-13	-1.0246289E-11	8.1484929E+04	1.0184735E+00	-5.1269849E-15	7.9920373E+04	3.3840184E-15
130 24	2.5630732E-13	8.3117986E-11	8.0889116E+04	1.0178330E+00	-2.7749115E-14	7.9275399E+04	3.5922663E-13
130 25	7.0566748E-13	-1.5191567E-10	7.9159041E+04	1.0161839E+00	1.4574248E-14	7.7494611E+04	3.3204854E-13
130 26	-3.7482058E-13	-4.4[illegible]1435E-11	8.0505504E+04	1.0143055E+00	2.4635318E-14	8.1542360E+04	-9.2323525E-13
130 27	-6.4182234E-13	1.6475934E-10	5.5636633E+04	1.0117116E+00	-2.2396888E-14	5.4992583E+04	3.5122692E-13
130 28	4.7743367E-13	-8.4840588E-11	3.8647897E+04	1.0083942E+00	-1.5992196E-14	3.8325387E+04	3.4612184E-13
130 29	-1.1888284E-13	-1.5688984E-11	2.0154167E+04	1.0048061E+00	8.4487511E-15	2.0057767E+04	1.4078914E-13
130 30	-2.2838664E-13	4.4382837E-11	8.2864531E+03	1.0020885E+00	8.0477025E-15	8.2733513E+03	-8.5579464E-14
130 31	8.5875743E-14	1.9237742E-11	1.5774118E+03	1.0011893E+00	-2.5111932E-15	1.5756632E+03	2.3095986E-13
130 32	2.2888196E-13	-2.1288918E-11	3.9160929E+03	1.0013840E+00	-8.1652905E-16	3.9106805E+03	3.1266917E-13
130 33	7.0631397E-14	5.9487807E-11	5.1714124E+03	1.0013725E+00	8.4731957E-15	5.1643242E+03	-7.5354518E-13
130 34	4.9863886E-13	-3.9797781E-11	2.2517126E+03	1.0006182E+00	7.6982119E-18	2.2503214E+03	-1.7053040E-13
130 35	2.0875719E-13	-7.3139628E-12	7.8969064E+02	1.0001781E+00	-4.3644040E-15	7.8955003E+02	8.3517229E-14
130 36	1.5814852E-13	-3.6759289E-11	6.1808351E+02	1.0001383E+00	-9.2645637E-17	6.1799805E+02	-7.4606514E-14
130 37	-1.8122562E-13	1.4347884E-11	-5.9712294E+02	9.9988786E-01	9.1802378E-15	-5.9718981E+02	-3.6943691E-13
130 38	0.						

TIME STEP 130 TIME 6.5000000E-05. SUBDIVISIONS OF TIME INCREMENT IN STRESS

(MATIN,K)

N	MP	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
1		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6		0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIME STEP 130 TIME .6500000E-04

SURFACE POWER VECTOR COMPONENTS

N	SNX(N)	SNR(N)	SNP(N)	SNRP(N)
4	.1000000000000000E+01	.7384779466444824E-15	.1000000000000000E+01	.6386724565084785E-15
5	.1000000000000000E+01	.5467846746864526E-15	.1000000000000000E+01	.4545230643027408E-15
6	.1000000000000000E+01	.3954536928154804E-15	.1000000000000000E+01	.3377836960590385E-15
7	.1000000000000000E+01	.2334141230521123E-15	.1000000000000000E+01	.1992725197248954E-15
8	.1000000000000000E+01	.7797553955414781E-16	.1000000000000000E+01	.7449063738866655E-16
9	.1000000000000000E+01	-.2871415845964351E-16	.1000000000000000E+01	-.8650622351031249E-16
10	.1000000000000000E+01	-.1414622322584454E-15	.1000000000000000E+01	-.2419137048302984E-15

11	.1000000000000000E+01	-.1840100762669015E-17	.1000000000000000E+01	.2496511939430835E-15
12	.1000000000000000E+01	.1434477254081701E-15	.1000000000000000E+01	.3447854217424421E-16
13	.1000000000000000E+01	.1809106960283168E-16	.1000000000000000E+01	.1497094404040323E-17
14	.1000000000000000E+01	-.7145061309630900E-16	.1000000000000000E+01	-.1448465038289539E-15
15	.1000000000000000E+01	-.2974034293557156E-16	.1000000000000000E+01	.8575595220101459E-16
16	.1000000000000000E+01	.2704412856734097E-16	.1000000000000000E+01	-.3180670724651101E-16
17	.1000000000000000E+01	-.4763722523842259E-16	.1000000000000000E+01	-.6349406363074950E-16
18	.1000000000000000E+01	-.3591877995726166E-17	.1000000000000000E+01	.5638544695446654E-16
19	.1000000000000000E+01	-.6796784560841311E-16	.1000000000000000E+01	-.1924449025513919E-15
20	.1000000000000000E+01	.2800363662199280E-16	.1000000000000000E+01	.2486106380341598E-15
21	.1000000000000000E+01	.1985373573011309E-15	.1000000000000000E+01	.1484280840424281E-15
22	.1000000000000000E+01	.5876271067320724E-16	.1000000000000000E+01	-.3094287098608377E-16
23	.1000000000000000E+01	-.1265806566975108E-15	.1000000000000000E+01	-.2722442271114370E-15
24	.1000000000000000E+01	-.9004064676119041E-16	.1000000000000000E+01	.4720452381624071E-16
25	.1000000000000000E+01	.1074884323123568E-15	.1000000000000000E+01	.1728252905553346E-15
26	.1000000000000000E+01	.3725581591105176E-17	.1000000000000000E+01	-.1655306394347867E-15
27	.1000000000000000E+01	-.1712543907607054E-15	.1000000000000000E+01	-.1769854747773809E-15
28	.1000000000000000E+01	-.1224139647976917E-16	.1000000000000000E+01	.1527734460087863E-15
29	.1000000000000000E+01	.1008832694611027E-15	.1000000000000000E+01	.4890040847063831E-16
30	.1000000000000000E+01	.4117063451293737E-16	.1000000000000000E+01	.3343038611333514E-16
31	.1000000000000000E+01	-.1688396318043852E-17	.1000000000000000E+01	-.3782434990123824E-16
32	.1000000000000000E+01	.1604425398703247E-16	.1000000000000000E+01	.6890560490814323E-16
33	.1000000000000000E+01	.7181279623862398E-16	.1000000000000000E+01	-.2528020264067721E-16
34	.1000000000000000E+01	-.3942592793747630E-16	.1000000000000000E+01	-.5357690397390391E-16
35	.1000000000000000E+01	-.3947604274704780E-16	.1000000000000000E+01	-.2537207930914419E-16
36	.1000000000000000E+01	-.8679980346598985E-17	.1000000000000000E+01	.8012450843123451E-17
37	.1000000000000000E+01	-.8027438936710431E-17	.1000000000000000E+01	-.2406933718042657E-16
38	.1000000000000000E+01	-.4011324426528629E-16	-.1000000000000000E+01	0.

SURFACE STRAINS, UPPER/LOWER, (EC1,EC2,E11,E22)

4	0.	3.6958367811407E-01	0.	4.3787972567821E-01
	0.	3.6958367811407E-01	0.	4.3787972567821E-01
5	0.	3.6694716018415E-01	0.	4.3427226936775E-01
	0.	3.6694716018415E-01	0.	4.3427226936775E-01
6	0.	3.6419972144017E-01	0.	4.3051975768097E-01
	0.	3.6419972144017E-01	0.	4.3051975768097E-01
7	0.	3.6125348591405E-01	0.	4.2650552645656E-01
	0.	3.6125348591405E-01	0.	4.2650552645656E-01
8	0.	3.3195384296331E-01	0.	3.8705051989236E-01
	0.	3.3195384296331E-01	0.	3.8705051989236E-01
9	0.	2.5317441751854E-01	0.	2.8522306036147E-01
	0.	2.5317441751854E-01	0.	2.8522306036147E-01
10	0.	1.6008328514365E-01	0.	1.7289661423483E-01
	0.	1.6008328514365E-01	0.	1.7289661423483E-01
11	0.	9.3235484485199E-02	0.	9.7581912268789E-02
	0.	9.3235484485199E-02	0.	9.7581912268789E-02
12	0.	5.4058393936828E-02	0.	5.5519540914347E-02
	0.	5.4058393936828E-02	0.	5.5519540914347E-02
13	0.	3.3978668703283E-02	0.	3.4547671869356E-02
	0.	3.3978668703283E-02	0.	3.4547671869356E-02
14	0.	2.4344379540437E-02	0.	2.4640703940041E-02
	0.	2.4344379540437E-02	0.	2.4640703940041E-02
15	0.	1.9408707690307E-02	0.	1.9597056657407E-02
	0.	1.9408707690307E-02	0.	1.9597056657407E-02
16	0.	1.6418614809184E-02	0.	1.6545268217593E-02
	0.	1.6418614809184E-02	0.	1.6545268217592E-02
17	0.	1.4365837497998E-02	0.	1.4469026141509E-02
	0.	1.4365837497998E-02	0.	1.4469026141510E-02
18	0.	1.2888257904216E-02	0.	1.2971311900121E-02
	0.	1.2888257904216E-02	0.	1.2971311900120E-02

19	0.	1.17500868369627E-02	0.	1.18191191073010E-02
	0.	1.17500868369627E-02	0.	1.18191191073008E-02
20	0.	1.08652913446679E-02	0.	1.09243186226683E-02
	0.	1.08652913446679E-02	0.	1.09243186226683E-02
21	0.	1.0[illegible]51026283 0E-02	0.	1.02535456608664E-02
	0.	1.02015102628390E-02	0.	1.02535456608664E-02
22	0.	9.60292563621590E-03	0.	9.73990203990087E-03
	0.	9.60292563621590E-03	0.	9.73990203990087E-03
23	0.	9.33053047803168E-03	0.	9.37405987753448E-03
	0.	9.33053047803168E-03	0.	9.37405987753448E-03
24	0.	9.03579790892918E-03	0.	9.07662073085190E-03
	0.	9.03579790892918E-03	0.	9.07662073085190E-03
25	0.	8.46837296840168E-03	0.	8.50422963876548E-03
	0.	8.46837296840168E-03	0.	8.50422963876548E-03
26	0.	7.59351627633018E-03	0.	7.67234707210499E-03
	0.	7.59351627633018E-03	0.	7.67234707210502E-03
27	0.	6.48324651902508E-03	0.	6.50426276173998E-03
	0.	6.48324651902508E-03	0.	6.50426276174027E-03
28	0.	5.01387546056268E-03	0.	5.02644493413448E-03
	0.	5.01387546056268E-03	0.	5.02644493413418E-03
29	0.	3.29465534784398E-03	0.	3.30008272477598E-03
	0.	3.29465534784398E-03	0.	3.30008272477598E-03
30	0.	1.72217531205388E-03	0.	1.72365825595748E-03
	0.	1.72217531205388E-03	0.	1.72365825595738E-03
31	0.	7.99121713434668E-04	0.	7.99441011192738E-04
	0.	7.99121713434668E-04	0.	7.99441011192668E-04
32	0.	6.23127863534028E-04	0.	6.23322007706108E-04
	0.	6.23127863534028E-04	0.	6.23322007706058E-04
33	0.	6.88896364401618E-04	0.	6.89133653500108E-04
	0.	6.88896364401618E-04	0.	6.89133653500228E-04
34	0.	4.97561658995958E-04	0.	4.97685442794608E-04
	0.	4.97561658995958E-04	0.	4.97685442794558E-04
35	0.	1.99057815564178E-04	0.	1.99077627573218E-04
	0.	1.99057815564178E-04	0.	1.99077627573198E-04
36	0.	7.90889373050168E-05	0.	7.90920648373298E-05
	0.	7.90889373050168E-05	0.	7.90920648373108E-05
37	0.	6.53363427005678E-06	0.	6.53365561846518E-06
	0.	6.53363427005678E-06	0.	6.53365561848168E-06
38	0.	-5.60732205212538E-05	0.	-5.60716484221827E-05
	0.	-5.60732205212538E-05	0.	-5.60716484220538E-05

SURFACE STRAINS, UPPER/LOWER, (EC1,EC2,E11,E22)

TIME STEP 130 TIME= .65000000E-04 KINETIC= .60262907E+05 ELASTIC= .13686924E+03 PLASTIC= .57101343E+04
TOTAL ENERGY= .66109990E+05

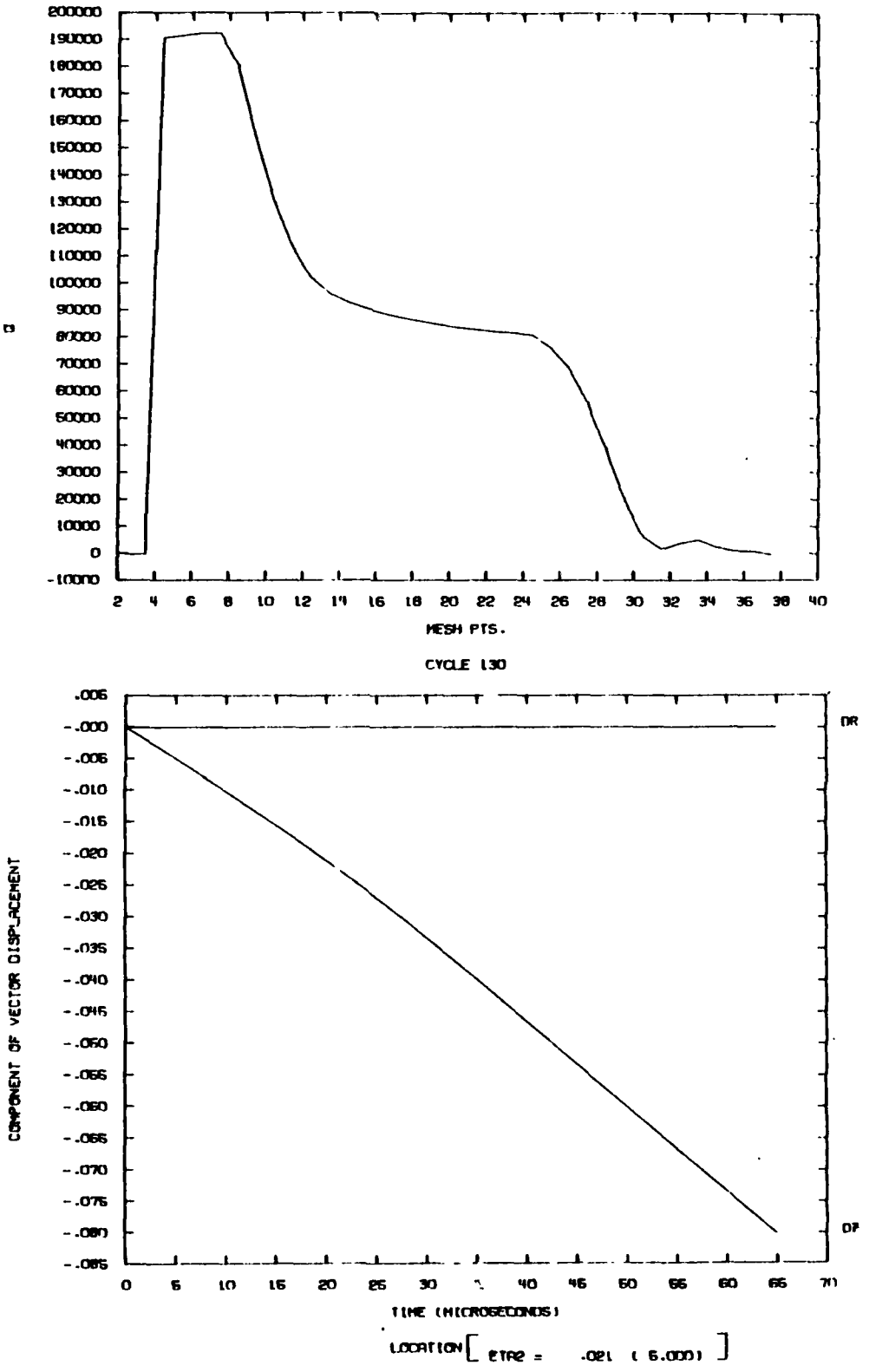


Figure E.2.1. Plot of axial force, Q, at cycle 130 and the displacement history at mesh point 5.

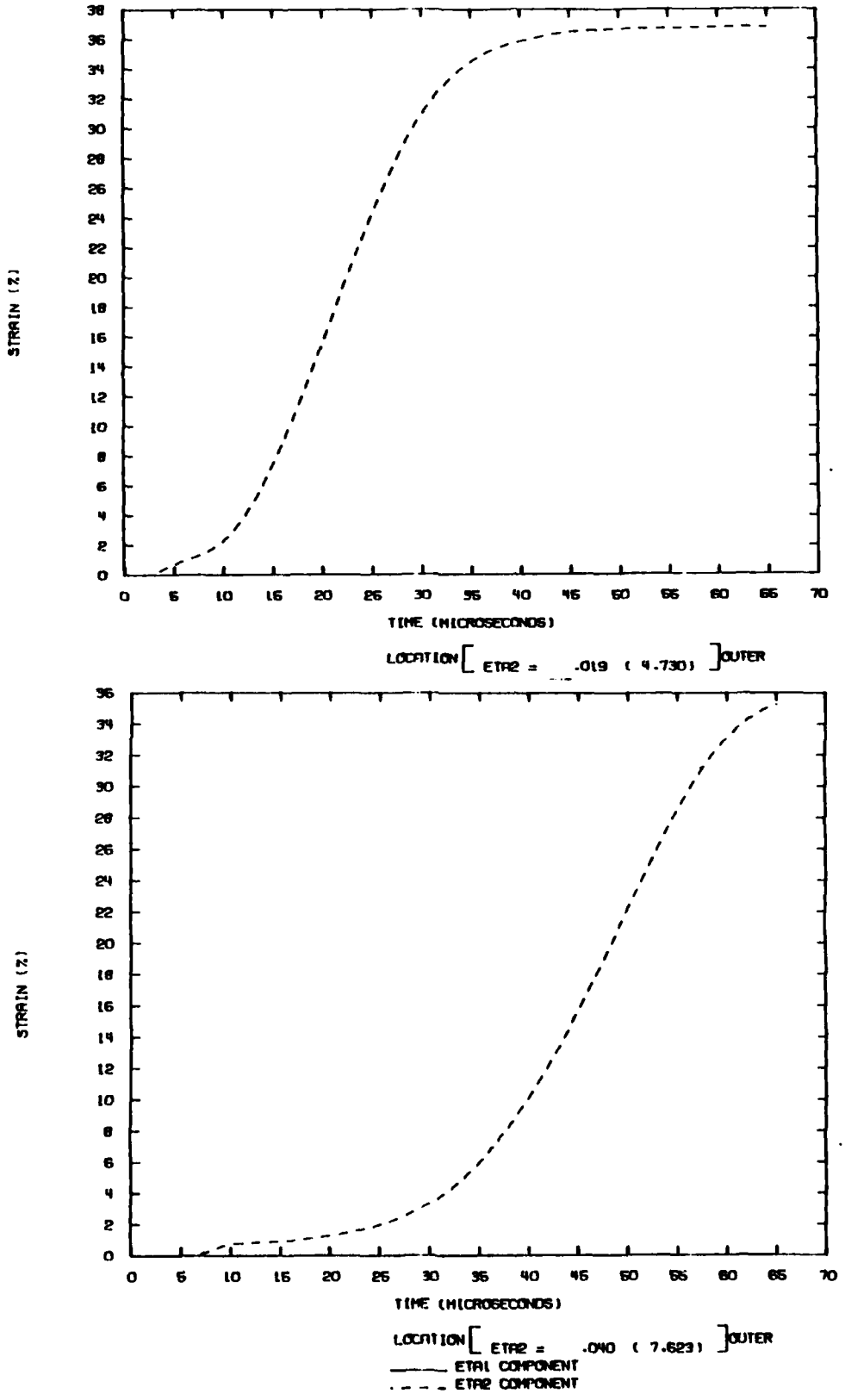


Figure E.2.2. Strain plots at 19.26 mm and 39.67 mm.

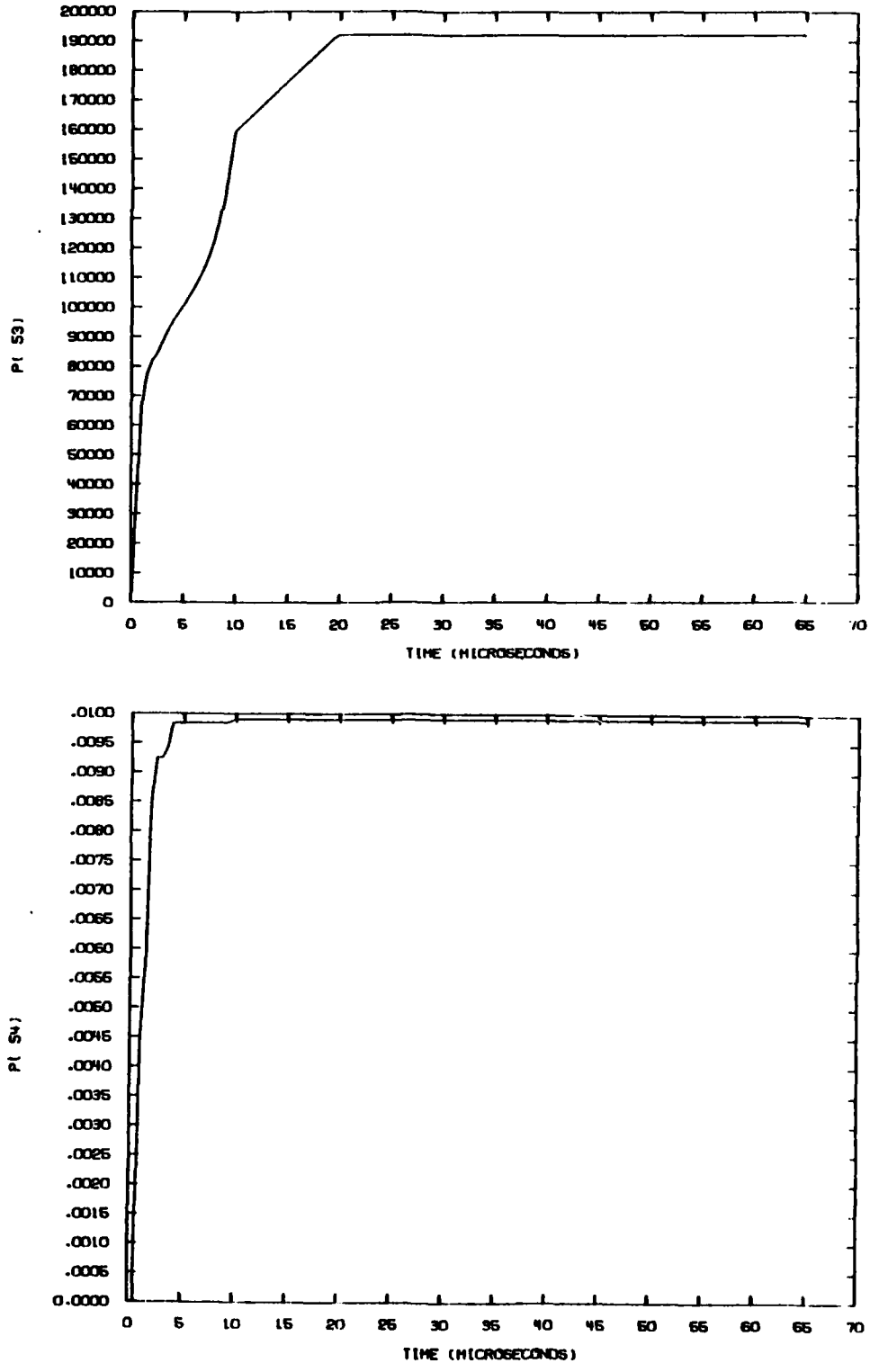


Figure E.2.3. Plots of force on end1, P(53), and erosion at end1, P(54), as functions of time.

APPENDIX F

SOME ALTERNATE USER SUBROUTINES

UPDATE input cards for alternate versions of the user subroutines INGEOM, PRESS, and ENDFRC are listed in this appendix. Other versions of the user subroutines are listed in Appendices C and E. Descriptions for user subroutines are given in Section 4.

The following are listed in the order given:

PRESS for Two Phase Pressure Decay in a Cylinder. This is described in Section 4.2.3.

INGEOM for an axisymmetric cylinder. This is described in Section 4.1.2.

ENDFRC for one or both ends free. This is described in Section 4.5.1.

```
*D  PRESS,4,PRESS,14
C         PRESS SUBROUTINE, 3/10/76.  TABULAR DATA.
C         POSITION ZSI = 0 CORRESPONDS TO THE 1=D REPSIL POINT N=2.
C         THIS PRESS SET UP FOR A SYMMETRIC AXISYMMETRIC CYLINDER
      DIMENSION ZSI(50),TSI(50),PSI(50),TS(103),PS(103),TSH(103),
     1 PSH(103),PID(6)
      DATA IPRESS/0/
      IF(IPRESS .GT. 0)GOTO 30
      IPRESS =1
      READ(5,100) PBAR,BETA,(PID(I),I=1,6)
      READ(5,105)CONCZ,CONCT,CONCP,TDIF
      WRITE(6,102) PBAR,BETA,(PID(I),I=1,6),TDIF,CONCZ,CONCT,CONCP
      READ(5,103)ALPHAS,NOUSE,ISM                                        12/2/75
      READ(5,101)(ZSI(I),TSI(I),PSI(I),I=1,ISM)
      WRITE(6,104) ALPHAS,(ZSI(I),TSI(I),PSI(I),I=1,ISM)                 12/2/75
C         CONVERT DATA TO Z=IN. T=SEC. P=LB/SQIN.                        12/2/75
      DO 2 I=1,ISM
      ZSI(I) = CONCZ*ZSI(I)
      TSI(I) = CONCT*(TSI(I)-TDIF)
      PSI(I)=CONCP*PSI(I)                                                12/2/75
    2 CONTINUE
      WRITE(6,106)(ZSI(I),TSI(I),PSI(I),I=1,ISM)
C         INITIATE DATA FOR COMPUTING THE PRESSURE
      DO 28 N=N1V,N2V
      DEN = FLOAT(N-2)*DETA2
      CALL DVDINT(DEN,TS(N),ZSI,TSI,ISM,2)
      CALL DVDINT(DEN,PS(N),ZSI,PSI,ISM,2)
      TSH(N) = TS(N) - ALOG(PBAR/PS(N))/ALPHAS
      PS(N) = PS(N)*EXP(ALPHAS*TS(N))
      PSH(N) = PBAR*EXP(BETA*TSH(N))
   28 CONTINUE
      WRITE(6,111)(Z(N),TS(N),PS(N),TSH(N),PSH(N),N=N1V,N2V)
C         COMPUTE THE PRESSURE
   30 EALPHT = EXP(-ALPHAS*TIME)
      EBETAT = EXP(-BETA*TIME)
      DO 39 N=N1V,N2V
      IF(TIME .LT. TS(N))GOTO 40
      IF(TIME .GT. TSH(N))GOTO 31
      P(N) = -EALPHT*PS(N)
      GOTO 39
   31 P(N) = -EBETAT*PSH(N)
   39 CONTINUE
   40 RETURN
  100 FORMAT(2E10.3,6A10)
  101 FORMAT(3E10.3)
  105 FORMAT(4E10.3)
  102 FORMAT(// 3X,'PRESS SUBROUTINE FOR 1-D STRUCTURE,3/10/76.'
     1 /' FROM TABLES OF ARRIVAL TIME AND PEAK PRESSURE VS POSITION.'/
     2 ' POSITION = 0 CORRESPONDS TO N=2 OF 1-D REPSIL.'/
     3 ' THE QUASI-STATIC PRESSURE IS',1PE15.7,5X,'WITH EXPONENTIAL '
     4 ' DAMPING COEFFICIENT =',E15.7/10X,6A10/
     * ' INPUT TIME, TSI, REDUCED BY TDIF=',  ,E15.7/
     5 ' CONVERSION COEFFICIENTS FOR Z, T, AND P ARE ',3E15.7)
  103 FORMAT(2E10.3,I10)
  104 FORMAT(//20X,'INPUT DATA              '/' ALPHAS=',1PE15.7,
     1                    //7X,'ZSI',12X,'TSI',12X,'PSI'/(3E15.7))
  106 FORMAT(//5X,'TABULAR DATA AFTER CONVERSION'/
     1 7X,'ZSI',12X,'TSI',12X,'PSI'/(1PE3E15.7))
  111 FORMAT(//5X'FINAL DATA              '/8X'Z'14X'TS'13X'PS'13X
     1 'TSH'12X'PSH'///(1P5E15.7))
      END
```

```
*D  INGEOM,7,INGEOM,47
*CALL MAIN
C
C  INGEOM FOR AXISYMMETRIC CYLINDER.  12/ 2/75.
C
C        SET PROGRAM TO USE GAUSSIAN INTEGRATION
      IGAUSS = 1
      READ(5,100)CYLL,RADIUS
      WRITE(6,101)CYLL,RADIUS
      DETA1 = 2.0*3.14159265358979*RADIUS
C        DETA1 = 2*PI*RADIUS GIVES ENERGY FOR ENTIRE CYLINDER
      DETA2 = CYLL/FLOAT(NMESH)
      IB = 0
C        *****     EVALUATE R(N) AND Z(N), N=N1B,N2B    *****
      DO 10 N=2,N2B
      R(N)=RADIUS
      Z(N)=FLOAT(N-2)*DETA2
   10 CONTINUE
      RETURN
  100 FORMAT(2E12.6)
  101 FORMAT(//' INGEOM FOR CYLINDER 12/3/75.  LENGTH ='1PE15.6,5X,
     1  ' RADIUS ='E15.6/)
      END
```

```
*AF
*DECK  ENDFRC                                                          ENDFRC
      SUBROUTINE ENDFRC                                                ENDFRC
*CALL MAIN
C                                                                      ENDFRC
C        THE APL FRC CHANGE TO (1-D) REPSIL FOR BEAMS REQUIRES A USER  ENDFRC
C        SUBROUTINE ENDFRC TO SUPPLY FORCES AND MOMENTS AT BOTH ENDS.  ENDFRC
C        FOR AN INITIALLY STRAIGHT BEAM, R(N)=0.0, Z(N)=(N-2)DETA2.    ENDFRC
C        FFZ1>0 DECREASES DZ(2).  EFR1>0 DECREASES DR(2)               ENDFRC
C        FFZ2>0 INCREASES DZ(N2B), EFR2>0 INCREASES DR(N2B)            ENDFRC
C        FM1>0 DECREASES DR(2)      AND INCREASES DR(3)                ENDFRC
C        FM2>0 INCREASES DR(N2B-1) AND DECREASES DR(N2B)               ENDFRC
C                                                                      ENDFRC
C        THIS VERSION IS FOR A BEAM WITH ONE, TWO, OR NO FREE ENDS.    ENDFRC
      DATA IENDFR/0/                                                   ENDFRC
      IF(IENDFR .GT. 0)RETURN                                          ENDFRC
      IENDFR = 1                                                       ENDFRC
      FFR1 = 0.0                                                       ENDFRC
      FFZ1 = 0.0                                                       ENDFRC
      FM1  = 0.0                                                       ENDFRC
      FFR2 = 0.0                                                       ENDFRC
      FFZ2 = 0.0                                                       ENDFRC
      FM2  = 0.0                                                       ENDFRC
      RETURN                                                           ENDFRC
      END                                                              ENDFRC
```

```
*ID  STREN
*I  RPSLID.5
C         NEW STRAIN ENERGY COMPUTATION FOR A BEAM.  1/20/76                STREN
*I  RMSTRS.7
C         NEW STRAIN ENERGY COMPUTATION  1/20/76                            STREN
      DIMENSION SST(5),DEST(5),LST(5)                                       STREN
*I  RMSTRS.33
C         NEW STRAIN ENERGY COMPUTATION  1/20/76                            STREN
      IF(IP .EQ. 2)GOTO 803                                                 STREN
      SST(J) = SIG22                                                        STREN
      LST(J) = J                                                            STREN
*D  RMSTRS.36
C         NEW STRAIN ENERGY COMPUTATION  1/20/76                            STREN
*I  RMSTRS.39
C         NEW STRAIN ENERGY COMPUTATION  1/20/76                            STREN
      IF(IP .EQ. 2)GOTO 35                                                  STREN
      IF(NSFL .GT. 1)GOTO 10                                                STREN
      DSTREN =(SS22)**2                                                     STREN
      GOTO 30                                                               STREN
   10 IF(SS22 .LE. 0.0)GOTO 12                                              STREN
      SS22 = -SS22                                                          STREN
      DO 11 L=1,NSFL                                                        STREN
   11 SST(L) = -SST(L)                                                      STREN
   12 DO 13 L=1,NSFL                                                        STREN
   13 DEST(L) = (SIGZZ(L) - SST(L))/E                                       STREN
      LLAST=NSFL-1                                                          STREN
      DO 15 L=1,LLAST                                                       STREN
      JSTART = L+1                                                          STREN
      DO 14 J=JSTART,NSFL                                                   STREN
      IF(DEST(L) .LT. DEST(J))GOTO 14                                       STREN
      DSTEMP = DEST(L)                                                      STREN
      DEST(L) = DEST(J)                                                     STREN
      DEST(J) = DSTEMP                                                      STREN
      LSTEMP = LST(L)                                                       STREN
      LST(L) = LST(J)                                                       STREN
      LST(J) = LSTEMP                                                       STREN
   14 CONTINUE                                                              STREN
   15 CONTINUE                                                              STREN
      EM = 1.0                                                              STREN
      DSTREN = 0.0                                                          STREN
      DO 25 L=1,NSFL                                                        STREN
      ESTAR = EM*E                                                          STREN
      DEPSTR = -SS22/ESTAR                                                  STREN
      DSTREN = DSTREN + (SS22)**2/EM                                        STREN
      IF(DEPSTR .LE. DEST(L)) GOTO 30                                       STREN
C      CORNER AT DEST(L) IN UNLOADING CURVE                                 STREN
      SS22 = -ESTAR*(DEPSTR-DEST(L))                                        STREN
      DSTREN = DSTREN - (SS22)**2/EM                                        STREN
      L1 = LST(L)                                                           STREN
      EM = EM - WT(L1)                                                      STREN
      DESTL = DEST(L)                                                       STREN
      DO 20 J=L,NSFL                                                        STREN
      DEST(J) = DEST(J) - DESTL                                             STREN
   20 CONTINUE                                                              STREN
   25 CONTINUE                                                              STREN
   30 STREN = STREN + SPG*W(K)*DSTREN                                       STREN
   35 CONTINUE                                                              STREN
```

LIST OF SYMBOLS

[FORTRAN name in brackets]

a	determinant of surface metric [DA]
E	Young's modulus of elasticity [E]
h	shell thickness [THICKN]
$\underset{\sim}{i}_1, \underset{\sim}{i}_2, \underset{\sim}{i}_3$	orthonormal vector basis
$\underset{\sim}{\hat{M}}^{*\alpha\beta}$	Normal vectors of stress moment resultant (See Reference 1). $\underset{\sim}{\hat{M}}^{*\alpha\beta} = \hat{M}^{*\alpha\beta}\,\underset{\sim}{n} = a^{\frac{1}{2}}\,\hat{M}^{\alpha\beta}\,\underset{\sim}{n} = a^{\frac{1}{2}}(M^{\alpha\beta} + M^{\beta\alpha})\underset{\sim}{n}$.
$\underset{\sim}{\hat{N}}^{*\alpha\beta}$	Modified stress resultant tensor (See Reference 1).
n or N	index associating variables with the N'th mesh point (e.g. R(N)) [N]
$\underset{\sim}{n}$	unit normal vector to reference surface, $\underset{\sim}{n} = n_k\,\underset{\sim}{i}_2 + n_r\,\underset{\sim}{i}_3$
$\underset{\sim}{r}$	position vector to a point on the reference surface
R,Z	position components. $\underset{\sim}{r} = R\,\underset{\sim}{i}_3 + Z\,\underset{\sim}{i}_2$ [R(N),Z(N)]
t	time [TIME]
W_k	area or weight associated with ζ_k [W(K)]
Δt	time increment [DELTAT]
$\Delta\xi, \Delta\xi^2$	spacing of ξ^2 between mesh points [DETA2]
ξ	ξ is ξ^2
ξ^1, ξ^2	material (Lagrangian) coordinates (called η^1, η^2 in Reference 2)
ζ	normal distance from reference surface
ζ_k	ζ at k'th integration station [ZETA(K)]
$\zeta_\ell, (\zeta_u)$	ζ on lower (upper) surface of shell [ZL,(ZU)]
ν	Poisson's ratio [FNU]
ρ	mass density [RHO]

DISTRIBUTION LIST

No. of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
3	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS (2 cys) DRDAR-TDC Dover, NJ 07801
1	Commander US Army Armament Materiel and Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
1	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Blvd. St. Louis, MO 63120
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications Rsch and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35898
1	Commander US Army Tank Automotive Rsch and Development Command ATTN: DRDTA-UL Warren, MI 48090
1	Director US Army TRACOD Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range NM 88002
1	AFELM, The Rand Corporation ATTN: Library-D 1700 Main Street Santa Monica, CA 90406
1	Massachusetts Institute of Technology Aeroelastic & Structures Rsch Lab 77 Massachusetts Ave ATTN: Dr. E. A. Witmer Cambridge, MA 02139

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen

Cdr, USATECOM
ATTN: DRSTE-TO-F

Dir, USACSL, Bldg. E3516, EA
ATTN: DRDAR-CLB-PA

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number _____

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

_____

_____

_____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) _____

_____

_____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

_____

_____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) _____

_____

_____

_____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: _____

Telephone Number: _____

Organization Address: _____

_____

_____

END

FILMED